



TUGAS AKHIR - IF184802

PERANCANGAN DAN IMPLEMENTASI GAME PUZZLE SIMULASI PENGATURAN LALU LINTAS KOTA: TRAFFIC PANIC BERBASIS ANDROID

ACHMAD IBNU MALIK AL CHASNI
NRP 05111540000175

Dosen Pembimbing
Imam Kuswardayan, S.Kom, M.T.
Dini Adni Navastara, S.Kom., M.Sc.

DEPARTEMEN INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN KOMUNIKASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2018

[Halaman ini sengaja dikosongkan]



TUGAS AKHIR - IF184802

PERANCANGAN DAN IMPLEMENTASI GAME PUZZLE SIMULASI PENGATURAN LALU LINTAS KOTA: TRAFFIC PANIC BERBASIS ANDROID

ACHMAD IBNU MALIK AL CHASNI
NRP 05111540000175

Dosen Pembimbing
Imam Kuswardayan, S.Kom, M.T.
Dini Adni Navastara, S.Kom., M.Sc.

DEPARTEMEN INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN KOMUNIKASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2018

[Halaman ini sengaja dikosongkan]



FINAL PROJECT - IF184802

DESIGN AND IMPLEMENTATION OF ANDROID BASED CITY TRAFFIC PUZZLE SIMULATION GAME : TRAFFIC PANIC

ACHMAD IBNU MALIK AL CHASNI
NRP 05111540000175

Advisor
Imam Kuswardayan, S.Kom, M.T.
Dini Adni Navastara, S.Kom., M.Sc.

INFORMATICS DEPARTEMENT
FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2018

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

PERANCANGAN DAN IMPLEMENTASI GAME PUZZLE SIMULASI PENGATURAN LALU LINTAS KOTA: TRAFFIC PANIC BERBASIS ANDROID

TUGAS AKHIR

Diajukan Guna Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Interaksi Grafika dan Seni
Program Studi S-1 Departemen Informatika
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember

Oleh :

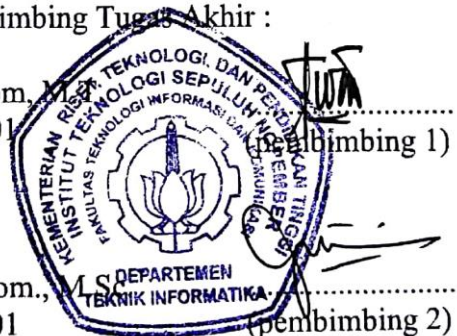
Achmad Ibnu Malik Al Chasni

NRP : 05111540000175

Disetujui oleh Dosen Pembimbing Tugas Akhir :

Imam Kuswardayan, S.Kom., M.Si

NIP: 197612152003121001



Dini Adni Navastara, S.Kom., M.Si

NIP: 198510172015042001

(pembimbing 2)

**SURABAYA
NOVEMBER 2018**

[Halaman ini sengaja dikosongkan]

**PERANCANGAN DAN IMPLEMENTASI GAME PUZZLE
SIMULASI PENGATURAN LALU LINTAS KOTA: TRAFFIC
PANIC BERBASIS ANDROID**

Nama Mahasiswa : ACHMAD IBNU MALIK AL CHASNI
NRP : 05111540000175
Departemen : Informatika FTIK-ITS
Dosen Pembimbing I : Imam Kuswardayan, S.Kom, M.T.
Dosen Pembimbing II : Dini Adni Navastara, S.Kom., M.Sc.

Abstrak

Kemajuan teknologi membuat banyak perubahan terutama dalam perkembangan *game* atau aplikasi permainan. Pengalaman bermain mulai berubah dari sesuatu yang statis menjadi dinamis. Contohnya dalam *game genre* simulasi yang menyediakan pengalaman mirip seperti keadaan dunia nyata yang disuguhkan dalam dunia permainan yang mengasyikan. *Genre* ini termasuk sangat populer karena memiliki nilai edukasi yang tinggi dan terbukti dapat menyampaikan nilai - nilai pembelajaran secara efektif tetapi tidak mengurangi unsur menyenangkan pada *game*.

Pada ‘Traffic Panic’ ini penulis menawarkan sebuah *game* simulasi lalu lintas kota yang didasari pada fakta bahwa kepadatan lalu lintas menjadi salah satu masalah utama yang terjadi di kota besar pada negara berkembang. Indonesia terutama kota Jakarta memiliki tingkat kepadatan lalu lintas yang sangat tinggi, menduduki peringkat ke-17 dengan indeks kemacetan mencapai 237,25. Kepadatan lalu lintas terjadi karena pertumbuhan infrastruktur jalan dan transportasi umum tidak berbanding lurus dengan pertumbuhan kendaraan serta akibat perilaku tidak tertib masyarakat dalam mematuhi peraturan lalu lintas. Perilaku

tersebut karena pendidikan mengenai pentingnya mematuhi aturan lalu lintas tidak diberikan pada pendidikan sekolah.

‘Traffic Panic’ merupakan *game puzzle-simulation* yang berfokus kepada penanganan kemacetan kota. Menggunakan keunggulan dari *genre* simulasi penulis berharap nilai edukasi akan tersampaikan secara lebih efektif. Dengan *genre* simulasi anak - anak akan berimajinasi seolah-olah benar - benar menata lalu lintas kota dengan berbagai kemungkinan kejadian. Hal ini akan melatih anak untuk berpikir cepat dalam menyelesaikan masalah sekaligus secara tidak langsung membuat mereka belajar memahami pentingnya mematuhi lampu lalu lintas untuk menghindari kemacetan.

Dari percobaan kepada 10 anak usia sekolah dasar, membuktikan *game* ini memberikan informasi baru mengenai peraturan perlalulintasan kepada target pengguna, menyampaikan simulasi peraturan perlalulintasan sesuai undang- undang nomor 22 tahun 2009 tentang lalu lintas dan angkutan jalan, dan menyampaikan unsur *fun* permainan dengan tingkat kesulitan yang terbukti sesuai kemampuan pengguna.

Kata kunci: lalu lintas, puzzle, simulasi

DESIGN AND IMPLEMENTATION OF ANDROID BASED CITY TRAFFIC PUZZLE SIMULATION GAME: TRAFFIC PANIC

Name : ACHMAD IBNU MALIK AL CHASNI
NRP : 0511540000175
Major : Informatics – FTIK ITS
Supervisor I : Imam Kuswardayan, S.Kom, M.T.
Supervisor II : Dini Adni Navastara, S.Kom., M.Sc.

Abstract

Technological advances have made many changes, especially in the development of games or game applications. The experience of playing starts to change from something static to dynamic. for example in simulation games that provide experiences similar to real-world conditions that are served in an exciting world of games. This genre is very popular because it has a high educational value and is proven to be able to convey learning values effectively but does not reduce the fun element in the game.

In this 'Traffic Panic', the author offers a city traffic simulation game based on the fact that traffic density is one of the main problems that occur in large cities in developing countries. Indonesia, especially the city of Jakarta, has a very high traffic density, ranks 17th with a congestion index reaching 237.25. traffic density occurs because the growth of road infrastructure and public transportation modes is not directly proportional to the growth of vehicles and due to the disorderly behavior of the community in complying with traffic regulations. Such behavior is because education about the importance of complying with traffic rules is not given to intra-school formal education

'Traffic Panic' is a puzzle-simulation game that focuses on handling city congestion. Using the advantages of the simulation genre the authors hope that the value of education will be delivered more effectively. With the genre of simulation, children will imagine as if they really organized city traffic with various possible events. This will train children to think quickly in solving problems while indirectly making them learn to understand the importance of obeying traffic lights to avoid congestion.

From testing with 10 target users, proving this game gives new information about channel balancing, and travel regulations according to law number 22 of 2009 concerning traffic and transportation to target users, and delivering fun games with difficulty levels that proved match with the user's ability .

Keywords: puzzle, simulation, traffic

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Alhamdulillahirabbil'alamin, segala puji bagi Allah SWT, yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul “PERANCANGAN DAN IMPLEMENTASI GAME PUZZLE SIMULASI PENGATURAN LALU LINTAS KOTA: TRAFFIC PANIC BERBASIS ANDROID”.

Pengerjaan Tugas Akhir ini merupakan suatu kesempatan yang sangat baik bagi penulis. Dengan pengerjaan tugas akhir ini penulis bisa mendapatkan ilmu lebih serta memanfaatkan semua ilmu yang telah didapatkan pada saat berkuliah di Departemen Informatika FTIK ITS.

Selesaiannya Tugas Akhir ini tidak lepas dari bantuan dan dukungan beberapa pihak. Sehingga pada kesempatan ini penulis mengucapkan syukur dan terima kasih kepada:

1. Allah SWT dan Nabi Muhammad SAW.
2. Orang tua, Kakak, Adik serta Saudara-saudara yang selalu mendoakan dan mendukung penulis.
3. Pak Imam Kuswardayan, S.Kom, M.T. selaku pembimbing I yang selalu memberikan arahan, motivasi dan bantuan sekaligus bimbingan kepada penulis selama pengerjaan Tugas Akhir.
4. Ibu Dini Adni Navastara, S.Kom., M.Sc. selaku pembimbing II yang juga telah sangat membantu, dan membimbing saat pengerjaan Tugas Akhir ini.

5. Sirria Panah Alam dan Irman Kurniawan yang tergabung bersama penulis dalam tim GG, yang telah membantu penulis selama pengerjaan Tugas Akhir.
6. Bapak dan Ibu Dosen Karyawan Teknik Informatika FTIf ITS yang telah memberikan ilmunya.
7. Cynthia Dewi, Hendry Wiranto, dan Pradipta Baskara yang menemani saat pengerjaan tugas akhir.
8. Steven, Yustian, Raca, Iyem, Yuuta, Pisjo, Bram, dan Brian yang terus menyemangati dalam pengerjaan tugas akhir.
9. Mas Syauki, mas Afiif, dan mas Galang yang membantu dalam penyusunan format buku TA.
10. Teman-teman angkatan 2015 yang telah membantu, berbagi ilmu, menjaga kebersamaan, dan memberi motivasi kepada penulis, kakak-kakak angkatan 2014, 2013, serta adik-adik angkatan 2016 dan 2017 yang membuat penulis untuk selalu belajar.
11. Laboratorium KCV tempat mengetjakan Tugas Akhir. Dan admin KCV yang sedikit membantu.
12. Serta semua pihak yang telah turut membantu penulis dalam menyelesaikan Tugas Akhir ini.

Penulis menyadari bahwa Tugas Akhir ini masih memiliki banyak kekurangan. Sehingga dengan kerendahan hati, penulis mengharapkan kritik dan saran dari pembaca untuk perbaikan ke depannya.

Surabaya, Desember 2018

DAFTAR ISI

Abstrak.....	vii
Abstract.....	ix
KATA PENGANTAR.....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR.....	xvii
DAFTAR TABEL	xix
DAFTAR KODE SUMBER	xxi
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Batasan Masalah.....	3
1.4. Tujuan.....	3
1.5. Manfaat.....	3
1.6. Metodologi Pembuatan Tugas Akhir	4
1.7. Sistematika Penulisan.....	5
BAB II TINJAUAN PUSTAKA	7
2.1. Rancang Bangun Perangkat Lunak	7
2.2. <i>Human Computer Interaction</i>	7
2.3. <i>Unity (Game Engine)</i>	7
2.4. Bahasa Pemrograman C#	8
2.5. <i>Game Puzzle</i>	8

2.6.	Android.....	8
2.7.	<i>Game</i> Yang Mirip Dengan ‘Traffic Panic’	8
2.8.	Algoritma <i>Shortest Path</i>	9
2.9.	Algoritma A*.....	10
2.10.	Algoritma IDA*.....	10
2.11.	Algoritma Dijkstra.....	10
BAB III ANALISIS DAN PERANCANGAN		11
3.1.	Analisis.....	11
3.1.1.	Deskripsi <i>Game</i> Traffic Panic.....	11
3.1.2.	Analisis <i>Gameplay</i> Aturan Permainan.....	11
3.1.3.	Analisis Komponen Permainan.....	14
3.2.	Perancangan	20
3.2.1.	Perancangan <i>Stage</i>	20
3.2.2.	Perancangan <i>Asset</i>	21
3.2.3.	Perancangan Algoritma <i>Shortest Path</i>	22
BAB IV IMPLEMENTASI.....		25
4.1.	Lingkungan Pengembangan Sistem	25
4.2.	Implementasi <i>Gameplay</i>	25
4.2.1.	Implementasi Algoritma <i>Shortest Path</i>	25
4.2.2.	Implementasi Pergerakan Kendaraan	30
4.2.3.	Implementasi Pengatur Lampu Lalu Lintas .	32
4.2.4.	Implementasi Pengontrol Bangunan	33
4.2.5.	Implementasi Perhitungan Skor.....	35
4.3.	Implementasi Antarmuka	36
4.3.1.	Implementasi <i>Main Menu</i>	36
4.3.2.	Implementasi <i>District Selection</i>	36
4.3.3.	Implementasi <i>Stage Selection</i>	37
4.3.4.	Implementasi <i>Gameplay Stage</i>	38

BAB V PENGUJIAN DAN EVALUASI	39
5.1. Lingkungan Pengujian.....	39
5.2. Pengujian Fungsionalitas.....	39
5.2.1. Skenario Pengujian Fungsionalitas	39
5.2.2. Hasil Pengujian Fungsionalitas.....	45
5.3. Pengujian Pengguna	46
5.3.1. Skenario Pengujian Pengguna.....	46
5.3.2. Hasil Pengujian Pengguna	46
5.4. Pengujian Algoritma <i>Shortest Path</i>	49
5.4.1. Skenario Pengujian Algoritma <i>Shortest Path</i> 49	
5.4.2. Hasil Pengujian Algoritma <i>Shortest Path</i>	49
5.5. Evaluasi Pengujian	50
BAB VI KESIMPULAN DAN SARAN	53
6.1. Kesimpulan.....	53
6.2. Saran.....	54
DAFTAR PUSTAKA.....	55
BIODATA PENULIS.....	57
LAMPIRAN A.....	59
LAMPIRAN B.....	69

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 2.1 <i>Game Traffic Command</i>	9
Gambar 3.1 Panic bar (atas) dan happiness bar (bawah) ...	12
Gambar 3.2 Rumah / <i>car spawner</i>	13
Gambar 3.3 Lampu lalu lintas pada persimpangan yang dapat di- <i>tap</i> untuk mengubah warnanya.	13
Gambar 3.4 Contoh bangunan spesial (gedung pemadam kebakaran).....	15
Gambar 3.5 Rumah terbakar dan truk pemadam kebakaran yang menuju ke lokasi kebakaran	15
Gambar 3.6 Perubahan pada mobil yang di- <i>upgrade</i>	18
Gambar 3.7 Flow diagram percobaan performa algoritma <i>shortest path</i>	22
Gambar 3.8 Diagram alir algoritma Dijkstra	23
Gambar 4.1 Implementasi tampilan <i>Main Menu</i>	36
Gambar 4.2 Implementasi tampilan <i>district selection</i>	37
Gambar 4.3 Implementasi tampilan <i>stage selection</i>	37
Gambar 4.4 Implementasi <i>Gameplay</i>	38
Gambar 5.1 Tampilan <i>Main Menu</i>	41
Gambar 5.2 Tampilan <i>District Selection</i>	41
Gambar 5.3 Tampilan <i>Stage Selection</i>	42
Gambar 5.4 Tampilan penggantian warna lampu lalu lintas	43
Gambar 5.5 Tampilan penggunakn <i>power-up</i>	43
Gambar 5.6 Kendaraan berjalan sesuai aturan permainan.	44
Gambar 5.7 Tampilan saat pemain menang.....	45
Gambar 5.8 Tampilan saat pemain kalah.....	45
Gambar A.6.1 <i>Stage 1-1</i>	59
Gambar A.6.2 <i>Stage 1-2</i>	59
Gambar A.6.3 <i>Stage 1-3</i>	60
Gambar A.6.4 <i>Stage 1-4</i>	60
Gambar A.6.5 <i>Stage 1-5</i>	61

Gambar A.6.6 <i>Stage</i> 1-6.....	61
Gambar A.6.7 <i>Stage</i> 1-7.....	62
Gambar A.6.8 <i>Stage</i> 1-8.....	62
Gambar A.6.9 <i>Stage</i> 1-9.....	63
Gambar A.6.10 <i>Stage</i> 1-10.....	63
Gambar A.6.11 <i>Stage</i> 1-11.....	64
Gambar A.6.12 <i>Stage</i> 1-12.....	64
Gambar A.6.13 <i>Stage</i> 1-13.....	65
Gambar A.6.14 <i>Stage</i> 1-14.....	65
Gambar A.6.15 <i>Stage</i> 1-15.....	66
Gambar A.6.16 <i>Stage</i> 1-16.....	66
Gambar A.6.17 <i>Stage</i> 1-17.....	67
Gambar A.6.18 <i>Stage</i> 1-18.....	67
Gambar A.6.19 <i>District</i> 2.....	68

DAFTAR TABEL

Tabel 3.1 Daftar Kendaraan.....	17
Tabel 3.2 Daftar Kondisi Kendaraan	18
Tabel 3.3 Daftar <i>Power-up</i>	20
Tabel 3.4 Konfigurasi <i>stage</i> game	21
Tabel 5.1 Lingkungan pengujian	39
Tabel 5.2 Pengujian aplikasi permainan	40
Tabel 5.3 Hasil pengujian fungsionalitas.....	46
Tabel 5.4 Hasil Pengujian Pengguna	47
Tabel 5.5 Hasil Kuisisioner Pengujian Pengguna	48
Tabel 5.6 Rangkuman Hasil Kuisisioner Pengujian Pengguna	48
Tabel 5.7 Hasil Pengujian Algoritma <i>Shortest Path</i>	49

[Halaman ini sengaja dikosongkan]

DAFTAR KODE SUMBER

Kode Sumber 4.1 Implementasi algoritma <i>Iterative Deepening A*</i>	26
Kode Sumber 4.2 Implementasi algoritma A*	28
Kode Sumber 4.3 Implementasi algoritma Dijkstra	29
Kode Sumber 4.4 Implementasi algoritma Dijkstra pada Unity	30
Kode Sumber 4.5 Pergerakan kendaraan	32
Kode Sumber 4.6 Pengatur Lampu Lalu Lintas.....	33
Kode Sumber 4.7 Pengontrol Bangunan.....	34
Kode Sumber 4.8 Perhitungan Skor.....	35

[Halaman ini sengaja dikosongkan]

BAB I PENDAHULUAN

Bab ini membahas garis besar penyusunan tugas akhir yang meliputi latar belakang, tujuan pembuatan, rumusan dan batasan permasalahan, metodologi penyusunan tugas akhir, dan sistematika penulisan.

1.1.Latar Belakang

Aplikasi permainan menjadi salah satu hiburan untuk berbagai macam kalangan. Perkembangannya juga sangat cepat dan hal tersebut membuat aplikasi terus berkembang mengikuti zaman. Berbagai perkembangan teknologi juga diterapkan di aplikasi permainan dengan tujuan untuk menambah pengalaman bermain pemain.

Salah satu genre *game* yang cukup banyak diminati adalah *genre simulation*. *Game* simulasi adalah *game* yang umumnya dirancang untuk mensimulasikan aktivitas dunia nyata, yang akhir-akhir ini makin sering digunakan dalam beberapa bidang seperti militer dan kedokteran [1]. Maka dari itu *game* simulasi harus memperhitungkan semua kejadian dalam kondisi sebenarnya. *genre* ini bertujuan untuk memberi pengalaman kepada pemain melalui permainan yang disimulasikan semirip-miripnya dengan kejadian nyata. Pengalaman yang disimulasikan biasanya adalah pengalaman tidak wajar yang jarang bisa dialami semua orang, seperti simulasi membangun kota, simulasi mengendarai pesawat dan simulasi lain-nya.

‘Traffic Panic’ merupakan *game puzzle simulation* yang bertemakan *government simulation* dimana pemain berperan sebagai pemerintah kota yang menata sebuah kota sehingga menjadi lebih tertib. Penataan kota disini berfokus kepada penataan *spot* keramaian, dan penempatan rambu lalu

lintas sehingga timbulnya ketertiban berlalu lintas dan tidak adanya kemacetan.

Tujuan dari pengerjaan Tugas Akhir ini adalah mampu menghasilkan aplikasi permainan yang tidak hanya menarik untuk dimainkan tetapi juga memberikan sebuah pelajaran mengenai rambu lalu lintas dan juga tata tertib berkendara kepada pemain, dengan cara mensimulasikan lalu lintas. Dengan pemain itu sendiri yang berperan sebagai pengatur tata lalu lintas kota, seperti mengatur lampu lalu lintas, rambu lalu lintas dan *traffic flow* di kota tersebut.

Sesuai tujuan, target pengguna yang optimal adalah anak- anak hingga remaja dimana rentang usia tersebut mereka mulai aktif sebagai pengguna jalan dan memerlukan edukasi mengenai peraturan lalu- lintas yang benar dan penanaman kebiasaan baik yang sesuai aturan undang-undang sedini mungkin.

1.2.Rumusan Masalah

Rumusan masalah yang diangkat dalam tugas akhir ini dapat dipaparkan sebagai berikut:

- a. Bagaimana merancang permainan simulasi pengaturan lalu lintas kota berdasarkan peraturan perundang- undangan di Indonesia sesuai undang undang nomor 22 tahun 2009 tentang lalu lintas dan angkutan jalan?
- b. Bagaimana merancang dan mengimplementasikan aturan main, skenario, dan tingkat kesulitan pada *game* ‘Traffic Panic’ berbasis Android?
- c. Bagaimana merancang dan mengimplementasikan *game* ‘Traffic Panic’ menggunakan *game engine* Unity?

1.3. Batasan Masalah

Permasalahan yang dibahas dalam tugas akhir memiliki beberapa batasan, yakni sebagai berikut:

- a. Peraturan lalu lintas dalam permainan simulasi menggunakan peraturan perundang-undangan di Indonesia sesuai undang undang nomor 22 tahun 2009 tentang lalu lintas dan angkutan jalan.
- b. Permainan dibuat menggunakan *game engine* Unity
- c. Untuk interaksinya, permainan menggunakan sentuhan jari tangan untuk memainkan permainan sebagai input dalam aplikasi permainan.

1.4. Tujuan

Tujuan dari pembuatan tugas akhir ini adalah membuat *game puzzle* simulasi lalu lintas kota: Traffic Panic berbasis android.

1.5. Manfaat

Manfaat dari hasil pembuatan tugas akhir ini diantaranya yaitu :

1. Mengetahui rancangan dan implementasi aturan main, skenario, dan tingkat kesulitan pada game 'Traffic Panic' berbasis Android.
2. Mengetahui rancangan dan implementasi game 'Traffic Panic' menggunakan *game engine* Unity
3. Mengetahui bagaimana menyampaikan pelajaran mengenai rambu lalu lintas dan juga tata tertib berkendara sesuai peraturan undang undang nomor 22 tentang lalu lintas dan angkutan jalan kepada pemain melalui *game* simulasi.
4. Meningkatkan pola pikir yang strategis bagi para pemain.
5. Memberikan hiburan bagi para pemain.

1.6. Metodologi Pembuatan Tugas Akhir

Pembuatan tugas akhir dilakukan menggunakan metodologi sebagai berikut:

A. Studi literatur

Tahap studi literatur merupakan tahap pembelajaran dan pengumpulan informasi yang digunakan untuk mengimplementasikan tugas akhir. Tahap ini diawali dengan pengumpulan literatur, diskusi, eksplorasi teknologi, dan pustaka, serta pemahaman dasar teori yang digunakan pada topik tugas akhir. Literatur-literatur yang dimaksud disebutkan sebagai berikut:

1. Unity
2. Bahasa pemrograman C#
3. Algoritma yang dibutuhkan dalam pembuatan *game* 'traffic panic'

B. Perancangan permainan

Pada tahap ini akan dilakukan analisa, perancangan, dan pendefinisian kebutuhan sistem untuk mengetahui permasalahan yang akan dihadapi pada tahap implementasi. Kemudian akan dijabarkan kebutuhan-kebutuhan tersebut ke dalam perancangan fitur sistem. Berikut langkah yang akan dilakukan perancangan proses aplikasi:

1. Perancangan *gameplay*
2. Perancangan menu
3. Perancangan data dan *asset* permainan
4. Pemilihan algoritma *shortest path*
5. Perencanaan dan pembuatan desain *level*

C. Implementasi dan pembuatan sistem

Aplikasi ini dibangun menggunakan *Game Engine Unity Personal Edition*, dengan bahasa pemrograman C#.

D. Pengujian dan Evaluasi

Tahap pengujian dan evaluasi berisi pengujian aplikasi dan evaluasi berdasarkan hasil pengujian. Pada tahap ini dilakukan pengujian dari fungsionalitas perangkat lunak, apakah sesuai dengan yang diharapkan serta tidak diharapkan terdapat *bug*. Pengujian terhadap *game* yang digunakan dilakukan dengan memastikan apakah *stage* dapat diselesaikan atau tidak.

E. Penyusunan laporan tugas akhir

Pada tahap ini dilakukan penyusunan laporan yang berisi dasar teori, dokumentasi dari perangkat lunak, dan hasil-hasil yang diperoleh selama pengerjaan tugas akhir.

1.7.Sistematika Penulisan

Buku Tugas Akhir ini terdiri atas beberapa bab yang tersusun secara sistematis, yaitu sebagai berikut.

1. BAB 1, Pendahuluan, menjelaskan latar belakang, batasan masalah, tujuan dari pembuatan tugas akhir ini serta metodologi yang digunakan selama penyusunan.
2. BAB 2, Tinjauan Pustaka, memaparkan hasil studi literatur yang digunakan sebagai dasar untuk menyelesaikan tugas akhir ini, terdiri atas deskripsi mengenai perancangan perangkat lunak, *human computer interaction*, Unity sebagai *game engine*, bahasa pemrograman C#, *game* simulasi, *game puzzle*, dan algoritma pencarian *shortest path*.
3. BAB 3, Analisa dan Perancangan sistem game yang dikembangkan. Pada tahap ini dijelaskan deskripsi dari

game ‘Traffic Panic’ dan dianalisa bagaimana *gameplay* dari *game* ‘Traffic Panic’. Setelahnya dibahas mengenai bagaimana perancangan *gameplay*, perancangan *stage*, dan bagaimana algoritma pencarian *shortest path* diterapkan dan disesuaikan dengan aturan permainan Traffic Panic.

4. BAB 4, Bab ini membahas implementasi dari rancangan sistem yang dilakukan pada tahap perancangan. Penjelasan implementasi meliputi implementasi pembuatan aplikasi permainan dengan menerapkan algoritma pencarian *shortest path* yang disesuaikan dengan aturan permainan ‘Traffic Panic’.
5. BAB 5, Pengujian dan Evaluasi, pengujian dilakukan dengan menguji setiap *stage*. Setiap *stage* akan diuji coba dan diamati apakah *stage* dapat diselesaikan atau tidak dan memenuhi kriteria *variable* peningkatan tingkat kesulitan.
6. BAB 6, Kesimpulan dan Saran, berisi tentang kesimpulan yang didapat dari proses pembuatan tugas akhir beserta saran-saran untuk pengembangan selanjutnya.

BAB II

TINJAUAN PUSTAKA

Bab ini membahas teori-teori yang mendukung pembuatan tugas akhir. Teori yang mendukung tersebut adalah deskripsi mengenai perancangan perangkat lunak, *human computer interaction*, Unity sebagai *game engine*, bahasa pemrograman C#, *game puzzle*, dan *Shortest path*.

2.1.Rancang Bangun Perangkat Lunak

Rancang bangun perangkat lunak adalah suatu ilmu yang mempelajari proses pembuatan aplikasi yang melingkupi analisis permasalahan dan kebutuhan, perencanaan, analisis sistem, implementasi, serta pemeliharaan perangkat lunak termasuk perbaikan jika ditemukannya *bug* [2].

2.2. Human Computer Interaction

Human Computer Interaction, atau dalam bahasa Indonesia yaitu “Interaksi Manusia dan Komputer” merupakan ilmu yang mempelajari tentang interaksi antara manusia dan komputer. Merancang bagaimana sistem computer agar efektif, efisien, mudah dan menyenangkan agar masyarakat dapat menyadari manfaat perangkat komputasional [3].

2.3.Unity (Game Engine)

Unity merupakan sebuah *game engine* yang dikembangkan oleh *Unity Technologies*. Unity dapat menciptakan *game* ke dalam beberapa sistem operasi sekaligus. Antara lain: Windows Phone, Android, iOS, Windows 8, OSX, Tizen OS, Blackberry 10, Playstation 3, Playstation 4, XBOX, Oculus Rift dan sebagainya. *Game*

yang dapat dibuat dengan Unity ini bisa dalam bentuk 3D atau 2D [4].

2.4.Bahasa Pemrograman C#

C# (dibaca: c sharp) merupakan sebuah bahasa pemrograman yang berorientasi objek yang dikembangkan oleh Microsoft sebagai bagian dari kerangka .Net framework. Bahasa pemrograman ini memiliki susunan yang mendekati bahasa C++.

2.5.Game Puzzle

Puzzle adalah salah satu *genre* permainan yang memiliki beberapa turunan contohnya *maze*. *Game* dengan *genre puzzle* sendiri memiliki banyak model permainan yang terus berkembang mengikuti zaman. *Game puzzle* memiliki penyelesaian yang tak pasti dan tentunya menggunakan sejumlah waktu untuk menyelesaikannya [5].

2.6.Android

Android adalah sebuah sistem operasi berbasis kernel Linux yang awalnya dikembangkan oleh Android .Inc yang didukung Google finansial dan kemudian dibeli pada tahun 2005 hingga akhirnya diresmikan pada tahun 2007. Android dapat menyesuaikan spesifikasi di kelas *low-end* hingga *high-end*. Android memiliki nama unik disetiap versinya yang diambil dari nama-nama makanan dan diurutkan berdasarkan huruf alfabet.[6]

2.7.Game Yang Mirip Dengan ‘Traffic Panic’

Terdapat *game* yang memiliki *gameplay* yang mirip dengan game ‘Traffic Panic’ dari segi layout, tujuan permainan, dan pergerakan karakter, yaitu *game* Traffic Command. Traffic Command adalah *game puzzle* simulasi yang berjalan di platform PC.



Gambar 2.1 Game Traffic Command

Sumber : Game Traffic Command

Tujuan pemain dalam permainan Traffic Command adalah mengubah warna lampu lalu lintas supaya tidak terjadi tabrakan antar mobil. Mobil akan muncul di ujung jalan dengan kecepatan tertentu dan hanya berhenti saat lampu merah menyala. Pemain harus memastikan bahwa hanya ada satu jalan pada persimpangan yang berlampu hijau atau mobil akan saling bertabrakan. Pemain menang jika tidak terjadi tabrakan sama sekali hingga waktu habis, dan pemain kalah jika terjadi tabrakan.

2.8. Algoritma *Shortest Path*

Jarak terpendek merupakan bagian dari teori graf. Jika diberikan sebuah graf berbobot, masalah jarak terpendek adalah bagaimana mencari sebuah jalur pada graf yang meminimalkan jumlah bobot sisi pembentuk jalur tersebut. Algoritma jarak terpendek dapat diaplikasikan untuk mencari rute antara lokasi fisik secara otomatis [7]

2.9. Algoritma A*

Algoritma A* adalah algoritma pencarian rute terpendek (*shortest path*) yang merupakan perbaikan dari Algoritme BFS dengan memodifikasi fungsi heuristiknya untuk memberikan hasil yang optimal. Dimana menggabungkan fungsi heuristik $[h(n)]$ dan jarak sesungguhnya/*cost* $[g(n)]$. [8]

2.10. Algoritma IDA*

Algoritma IDA* adalah algoritma pencarian rute terpendek (*shortest path*) yang merupakan gabungan dari Algoritme IDBFS dan A* tanpa menggunakan *dynamic programming*. Sehingga memiliki konsumsi memory minimal tetapi memungkinkan melakukan double visit sehingga memakan lebih banyak waktu [9]

2.11. Algoritma Dijkstra

Algoritma Dijkstra adalah sebuah algoritme rakus (*greedy algorithm*) yang dipakai dalam memecahkan permasalahan jarak terpendek (*shortest path problem*) untuk sebuah graf berarah (*directed graph*) dengan bobot-bobot sisi (*edge weights*) yang bernilai tak-negatif. [10]

BAB III

ANALISIS DAN PERANCANGAN

Bab ini membahas tahap analisis permasalahan dan perancangan tugas akhir. Pada bagian awal akan dibahas mengenai *gameplay* atau aturan dari *game* ‘Traffic Panic’. Selajutnya dibahas mengenai penggunaan algoritma *shortest path* yang digunakan. Algoritma *shortest path* yang digunakan pada *game* Traffic Panic telah disesuaikan dengan aturan permainan agar dapat menghasilkan jalan terpendek dengan cepat tanpa membebani *device* yang digunakan.

3.1. Analisis

3.1.1. Deskripsi Game Traffic Panic

Traffic Panic yang dalam artian bahasa indonesia adalah kepanikan lalu lintas adalah sebuah *game* ber-genre simulasi *puzzle* dengan tema edukasi lalu lintas. Pada *game* ini pemain dihadapkan pada sebuah ruas kota, dengan persimpangan, rumah dan kendaraan yang melintas. Dalam *game* ini, pemain dapat mengatur laju kendaraan dengan mengubah warna lampu di setiap persimpangan. Pemain dapat menyelesaikan *stage* dengan menjaga lalu lintas tetap lancar.

3.1.2. Analisis Gameplay Aturan Permainan

‘Traffic Panic’ memiliki aturan yang cukup sederhana. Untuk awal mula permainan pemain akan dihadapkan pada sebuah ruas kota dengan persimpangan dan beberapa rumah.

Selama permainan, mobil- mobil akan muncul dari rumah dan ujung jalan menuju tujuan tertentu dan memadati jalan. Pemain dapat melakukan *tap* pada lampu lalu lintas untuk mengubah warna lampu tersebut. Mobil yang melintas

akan mematuhi lampu lalu lintas, dan berhenti saat merah. Dengan mengubah warna lampu lalu lintas, pemain harus mengatur strategi sehingga tidak terjadi kemacetan.

Dalam *game* Traffic Panic terdapat 2 *bar* yang mempengaruhi menang dan kalahnya pemain. *Bar* yang pertama adalah *happiness bar*. Jika tidak terjadi kemacetan, *bar* ini akan bertambah seiring berjalannya waktu. *Bar* yang kedua adalah *Panic Bar* *bar* ini akan bertambah jika terjadi kemacetan. Semakin parah kemacetan yang terjadi, semakin cepat *bar* ini bertambah penuh. Jika *happiness bar* terisi penuh sebelum *panic bar*, pemain dinyatakan menang. Sebaliknya, Jika *panic bar* terisi penuh sebelum *happiness bar*, pemain dinyatakan kalah.



Gambar 3.1 Panic bar (atas) dan happiness bar (bawah)

Berbagai macam kendaraan akan keluar dari rumah dan ujung jalan, memadati jalan hingga sampai pada tujuan tertentu dan hanya berhenti saat lampu merah atau jika terdapat kendaraan lain di depannya. kendaraan yang berhenti dan memadati jalan inilah yang memicu naiknya *panic bar*. Pemain dapat mencegah hal ini terjadi dengan melakukan *tapping* pada lampu lalu lintas di persimpangan jalan, agar kendaraan tersebut dapat melanjutkan perjalanan.



Gambar 3.2 Rumah / car spawner



Gambar 3.3 Lampu lalu lintas pada persimpangan yang dapat di-tap untuk mengubah warnanya.

Dalam setiap level, pemain harus mengatur lalu lintas agar tidak terjadi kemacetan dan mencegah panic bar bertambah, hingga happiness bar penuh.

3.1.3. Analisis Komponen Permainan

Ada beberapa komponen yang ada dalam aplikasi permainan 'Traffic Panic'. Komponen tersebut akan mempengaruhi jalannya permainan.

3.1.3.1.Susunan *Graph*

Traffic Panic memiliki susunan *stage* berupa *graph* tak kasat mata yang ditampilkan melalui rumah, jalan dan persimpangan jalan. *Graph* yang ada pada *game* ini berbeda tergantung dari *stage* yang dimainkan. *Graph* bersifat *dynamic directed weighted*, sesuai dengan jalan yang dapat dilewati oleh kendaraan.

3.1.3.2.Bangunan

Bangunan adalah objek dalam permainan. Bangunan pada Traffic Panic ada 2 macam. Yang pertama adalah bangunan biasa, yaitu bangunan tempat kendaraan keluar dan masuk. Tipe bangunan kedua adalah bangunan spesial. Yaitu bangunan yang mengeluarkan kendaraan spesial yang memiliki peranan tertentu dalam permainan. Bangunan tersebut adalah rumah sakit dan gedung pemadam kebakaran. Gedung pemadam kebakaran akan mengirimkan truk pemadam kebakaran ke rumah yang terbakar, dan rumah sakit akan mengirimkan ambulans ke rumah yang terserang wabah penyakit.



Gambar 3.4 Contoh bangunan spesial (gedung pemadam kebakaran)



Gambar 3.5 Rumah terbakar dan truk pemadam kebakaran yang menuju ke lokasi kebakaran

3.1.3.3.Kendaraan

Kendaraan adalah objek dalam permainan. Kendaraan menjadi *variable* yang dikendalikan dalam permainan dan penentu apakah pemain menang atau kalah. Semua kendaraan memiliki tingkah laku yang sama, yaitu berangkat dari 1 titik (bangunan atau ujung jalan) menuju titik tujuan (bangunan atau ujung jalan). Kendaraan dibagi menjadi 2 macam, yaitu kendaraan biasa dan kendaraan spesial.

Kendaraan spesial seperti yang disebut diatas, adalah kendaraan yang keluar dari bangunan spesial dan memiliki tujuan tertentu, seperti menyembuhkan orang sakit atau memadamkan kebakaran. Saat ini, dalam *game* Traffic Panic, terdapat 2 kendaraan spesial, yaitu ambulans dan truk pemadam kebakaran.

Sedangkan kendaraan biasa dibagi lagi menjadi 2, yaitu kendaraan yang dapat di-*upgrade* dan kendaraan yang tidak dapat di-*upgrade*. Kendaraan yang dapat di-*upgrade* ada 8 macam mobil, dengan warna yang berbeda dan kekuatan yang berbeda. Tetapi tidak semua kendaraan memiliki pengaruh positif, beberapa kendaraan juga memiliki efek negatif yang memberi tantangan tersendiri untuk pemain.

Untuk penjelasan setiap kendaraan dapat dilihat di tabel 3.1.

Tabel 3.1 Daftar Kendaraan

No	Gambar Kendaraan	Nama Kendaraan	Efek
1		<i>Blue car</i>	Dapat mengeluarkan <i>diamond</i>
2		<i>Indigo car</i>	Dapat memanipulasi waktu
3		<i>Green car</i>	Menambah <i>happiness bar</i> lebih cepat
4		<i>Gray car</i>	-
5		<i>Orange car</i>	Dapat mengeluarkan <i>power-up magic wand</i>
6		<i>Pink car</i>	Mengurangi kecepatan pertambahan kemacetan
7		<i>Purple car</i>	Dapat berubah menjadi <i>invisible</i>
8		<i>Red car</i>	Menghentikan <i>spawn</i> dari sebuah bangunan
9		<i>White car</i>	-
10		<i>Yellow car</i>	Dapat mengeluarkan <i>coin</i>
11		<i>Bus</i>	Akan melambat saat bertemu <i>bus stop sign</i>
12		<i>Dump truck</i>	Dapat menjatuhkan sampah
13		<i>Ambulance</i>	Menuju ke rumah yang sakit dan menyembuhkan penyakit
14		<i>Fire truck</i>	Menuju ke rumah yang kebakaran dan memadamkan api


Kendaraan yang *di-upgrade* akan memiliki perubahan tampilan. Pemain dapat melakukan *upgrade* kendaraan di *shop*. Mobil yang *di-upgrade* akan semakin membantu pemain dalam menyelesaikan permainan Untuk gambar *upgrade* kendaraan dapat dilihat di Gambar 3.6.



Gambar 3.6 Perubahan pada mobil yang di-upgrade

Kendaraan juga memiliki kondisi tertentu. Kondisi ini menambah keseruan dalam permainan. Daftar kondisi yang dapat dialami kendaraan dapat dilihat pada tabel 3.2.

Tabel 3.2 Daftar Kondisi Kendaraan

No	Gambar	Kondisi	Penjelasan
1		<i>Broke down</i>	Mobil mogok. Ditandai dengan munculnya asap hitam di atas mobil. Pemain harus melakukan <i>tapping</i> sebanyak 3-5 kali untuk memperbaikinya
2		<i>Slowed</i>	Mobil melambat. Ditandai dengan berkurangnya kecepatan mobil. Mobil melambat akibat <i>power-up Freeze</i>
3		<i>Sleepy driver</i>	Pengemudi mengantuk. Ditandai dengan <i>flashing light</i> disekitar mobil. Jika dibiarkan akan terjadi <i>crashing car</i> . Pemain harus melakukan <i>tap</i> untuk menyadarkan pengemudi
4		<i>Crashing car</i>	Mobil bertabrakan. Ditandai dengan munculnya suara tabrakan dan gambar “Bang” disertai asap. Terjadi ketika ada pengemudi ngantuk yang tetap dibiarkan.
5		<i>Teleport car</i>	Mobil teleport menuju tujuan akhir. Ditandai dengan menghilangnya mobil dan berubah menjadi butiran bintang. Terjadi ketika pemain menggunakan <i>magic wand power-up</i>
6		<i>Invisible car</i>	Mobil tidak terlihat. Ditandai dengan gelembung hitam mengelilingi mobil dan <i>opacity</i> 0,7. mobil akan menembus kemacetan dan melaju terus sampai tujuan. Terjadi ketika pemain menggunakan <i>power-up invisibility</i>

3.1.3.4.Lampu Lalu Lintas

Lampu lalu lintas adalah objek dalam permainan dan satu- satunya objek dalam permainan yang interaktif. Dengan men-*tap* lampu akan berubah warna. Hal ini menjadi inti dari permainan, dimana dengan mengubah warna lampu lalu lintas, pemain dapat mengatur pergerakan lalu lintas untuk menghindari kemacetan.

3.1.3.5.Panic Bar, Happiness Bar, Time, Coin, dan Scoring

Panic bar, happiness bar time adalah komponen permainan yang beragam setiap *stage*-nya. Jumlah *panic bar* dan *happiness bar* maksimal akan berbeda- beda setiap *stage*-nya. *Coin* yang ditampilkan pada pojok kanan atas layar adalah *coin* yang didapatkan pada *stage* ini, dan *time* adalah selang waktu *stage* dimainkan. *scoring* pada Traffic Panic bergantung pada 3 hal, yaitu *panic bar, happiness bar* dan *time*. Semakin penuh *happiness bar* semakin banyak skor yang didapat. Sebaliknya semakin penuh *panic bar* semakin sedikit skor yang didapat. Sedangkan semakin lama *time* juga semakin sedikit skor. Jadi semakin cepat sebuah *stage* diselesaikan, semakin banyak skor yang didapat.




Skor setiap *stage* yang didapat ini mempengaruhi 3 hal, yaitu jumlah bintang yang didapat pada *stage*, XP yang didapat, dan *difficulties multiplier* untuk level selanjutnya. Bintang yang didapat berfungsi untuk kepuasan atau pencapaian pribadi pemain. XP yang didapat adalah total skor setiap *level*, dan XP inilah yang akan masuk kedalam database *LeaderBoard*. Sedangkan *difficulties multiplier* adalah sebuah fitur Traffic Panic dengan penggunaan AI yang mengatur tingkat kesulitan permainan berdasarkan kemampuan pemain sehingga pemain tidak

merasa bosan karena terlalu mudah atau menyerah karena terlalu susah.

3.1.3.6. *Power Up*

Power-up adalah sebuah fitur dalam permainan guna membantu pemain mempermudah menyelesaikan sebuah *stage*. Terdapat 4 *power-up* dalam permainan Traffic Panic, yang masing masing memiliki fungsi yang berbeda. *Power-up* yang dimiliki pemain terbatas, tetapi dapat dibeli kembali di *shop*. Penjelasan mengenai *power-up* dapat dilihat pada tabel 3.3.

Tabel 3.3 Daftar *Power-up*

No	Gambar	Nama	Penjelasan
1		<i>Stop</i>	<i>Stop power-up</i> berfungsi untuk menghentikan bangunan selain bangunan spesial supaya tidak mengeluarkan kendaraan selama durasi tertentu.
2		<i>Magic Wand</i>	<i>Magic power-up</i> berfungsi untuk memindah semua kendaraan di jalan ke tujuan akhir, mengosongkan jalan sehingga bar kemacetan berkurang.
3		<i>Freeze</i>	<i>Freeze power-up</i> memperlambat waktu, mengakibatkan semua kendaraan berjalan lebih lambat.
4		<i>Invisible</i>	<i>Invisibility power-up</i> membuat 1 buah kendaraan berjalan bebas menembus kemacetan, dan tidak berhenti sampai menuju tujuan.

3.2. Perancangan

3.2.1. Perancangan *Stage*

‘Traffic Panic’ memiliki 30 *stage*, dengan *stage* pertama adalah *tutorial*. 18 *stage* pada *district* 1, dan 12 *stage* pada *district* 2. Perbedaan dari *district* 1 dan 2 terletak pada ukuran *map*. Sedangkan perbedaan antar *stage* ada pada jumlah persimpangan dan *poin* kebahagiaan dan kepanikan maksimal. Berikut pembagian format *stage* pada tiap *stage*:

Tabel 3.4 Konfigurasi *stage game*

No	<i>District - Stage</i>	Jumlah persimpangan	Maksimal kendaraan	Maksimal poin kebahagiaan	Maksimal poin kepanikan
1	1-1	1	7	1000	2000
2	1-2	1	7	800	800
3	1-3	1	7	800	750
4	1-4	1	7	850	700
5	1-5	1	8	850	700
6	1-6	1	8	850	650
7	1-7	2	8	800	750
8	1-8	1	11	900	700
9	1-9	2	10	800	700
10	1-10	3	11	800	750
11	1-11	1	14	800	750
12	1-12	1	16	900	950
13	1-13	2	17	800	600
14	1-14	2	16	900	950
15	1-15	2	20	800	700
16	1-16	2	19	1000	600
17	1-17	2	20	950	600
18	1-18	3	17	950	600
19	2-1	6	12	1000	850
20	2-2	6	12	1050	750
21	2-3	6	12	1150	700
22	2-4	6	13	1000	850
23	2-5	6	13	1050	750
24	2-6	6	13	1150	700
25	2-7	6	15	900	850
26	2-8	6	16	950	850
27	2-9	6	17	1000	850
28	2-10	6	18	1100	850
29	2-11	6	20	1150	900
30	2-12	6	22	1200	950

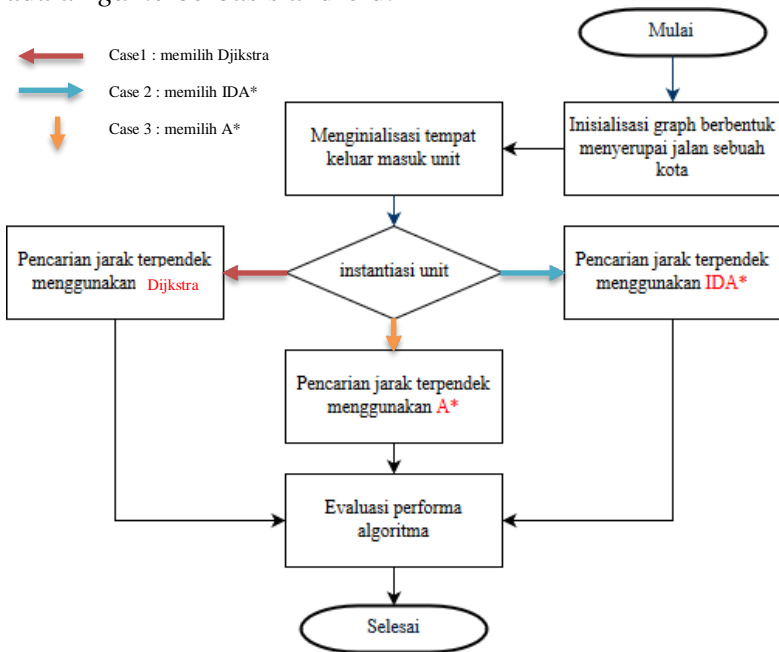
3.2.2. Perancangan *Asset*

Semua *asset* gambar merupakan *asset* internal penulis yang dibuat oleh pihak kedua dan sudah diberi izin

pemakaian *asset*. *Asset* suara didapatkan oleh penulis dari audioblocks.com melalui akun Lab IGS.

3.2.3. Perancangan Algoritma *Shortest Path*

Algoritma *Shortest Path* diperlukan pada *gameplay game* Traffic Panic pada bagian pencarian *path* yang dilalui setiap kendaraan. Algoritma *Shortest Path* yang dipilih harus sangat cepat karena algoritma tersebut akan dijalankan setiap ada kendaraan baru yang muncul. Selain itu juga tidak boleh terlalu berat atau memakan memori karena Traffic Panic adalah *game* berbasis android.



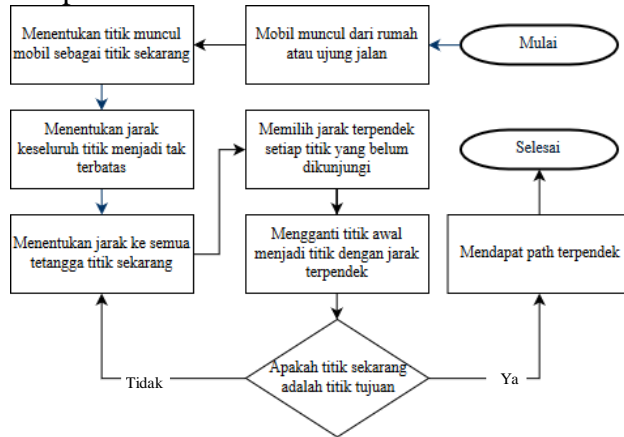
Gambar 3.7 Flow diagram percobaan performa algoritma *shortest path*

Setelah melakukan percobaan dengan menggunakan 3 algoritma berbeda, didapatkan algoritma dijkstra menjadi

algoritma paling efisien dengan akurasi 100%, waktu paling cepat.

3.2.3.1.Dijkstra

Dijkstra yang dipakai telah disesuaikan dengan peraturan permainan Traffic Panic.



Gambar 3.8 Diagram alir algorithma Dijkstra

Berdasarkan gambar tersebut, berikut adalah tahapan perancangan dan penerapan algoritma Dijkstra pada aplikasi permainan:

1. Pencarian jalan terpendek dimulai saat kendaraan muncul.
2. Setelah kendaraan muncul, jarak akhir ke setiap *node* di-*set* menjadi tak terbatas.
3. Menjadikan titik *spawn* sebagai titik sekarang.
4. Melakukan iterasi ke setiap tetangga titik sekarang untuk menentukan jarak akhir ke setiap tetangga.
5. Mengupdate jarak akhir jika jarak baru lebih pendek dari jarak sebelumnya.

6. Menandai titik sekarang menjadi titik yang telah dikunjungi dan mengganti titik sekarang menjadi titik dengan jarak akhir terpendek.
7. Mengulangi *step* 4-6 jika titik sekarang bukanlah titik yang dituju.
8. Jika titik sekarang adalah titik yang dituju, jarak dan *path* sudah didapatkan.

BAB IV IMPLEMENTASI

Bab ini membahas mengenai implementasi yang dilakukan berdasarkan rancangan yang telah dijabarkan pada bab sebelumnya. Implementasi yang dijelaskan adalah pembuatan *game* Traffic Panic dengan didukung algoritma *shortest path* yang efisien. *Game Engine* yang digunakan adalah Unity dengan bahasa pemrograman C#.

4.1. Lingkungan Pengembangan Sistem

Lingkungan pengembangan sistem yang digunakan untuk mengembangkan Tugas Akhir ini dilakukan pada lingkungan pengembangan sebagai berikut.

1. Sistem operasi Windows 10 Home 64 bit.
2. Unity Game Engine 2018.2.1
3. Visual Studio Code 1.25.1

4.2. Implementasi *Gameplay*

Implementasi dari semua fungsi diwujudkan dalam bentuk *code* dengan Bahasa pemrograman C#.

4.2.1. Implementasi Algoritma *Shortest Path*

Sesuai perancangan, percobaan dengan 3 algoritma *shortest path* A*, IDA* dan Dijkstra dengan implementasi algoritma sebagai berikut sesuai Kode Sumber 4.1. – Kode Sumber 4.3.

```

1.  public class Tree<K, V>
2.      where K : class, IComparable<K>
3.      where V : class
4.  {
5.      private Node<K, V> root;
6.      public V DepthLimitedSearch(Node<K, V> root, K goal, int depth)
7.      {
8.          if(depth == 0 && root.key == goal)
9.          {

```

```

10.         return root.value;
11.     }
12.     else if (depth > 0)
13.     {
14.         foreach (var child in root.children)
15.         {
16.             var result = DepthLimitedSearch(child, goal, depth-
17. 1);
18.             if(result != default(V))
19.                 return result;
20.         }
21.         return default(V);
22.     }
23.     else
24.         return default(V);
25.
26.     public V IterativeDeepeningSearch(K key, int depth)
27.     {
28.         for (int currentDepth = 0; currentDepth <= depth; currentDepth++)
29.         {
30.             var v = DepthLimitedSearch(root, key, currentDepth);
31.             if(v != default(V))
32.                 return v;
33.         }
34.         return default(V);
35.     }
36.
37.     private class Node<K, V>
38.     where K : class, IComparable<K>
39.     where V : class
40.     {
41.         public K key;
42.         public V value;
43.         public Node<K, V>[] children;
44.     }
45. }

```

Kode Sumber 4.1 Implementasi algoritma *Iterative Deepening A**

```

1. public List<Node> GetShortestPathAstar()
2. {
3.     foreach (var node in Map.Nodes)
4.         node.StraightLineDistanceToEnd = node.StraightLineDistanceTo(End);
5.     AstarSearch();
6.     var shortestPath = new List<Node>();
7.     shortestPath.Add(End);
8.     BuildShortestPath(shortestPath, End);
9.     shortestPath.Reverse();
10.    return shortestPath;
11. }
12.
13. private void AstarSearch()
14. {
15.     Start.MinCostToStart = 0;
16.     var prioQueue = new List<Node>();
17.     prioQueue.Add(Start);
18.     do {
19.         prioQueue = prioQueue.OrderBy(x => x.MinCostToStart + x.StraightLineDistanceToEnd).ToList();
20.         var node = prioQueue.First();
21.         prioQueue.Remove(node);
22.         NodeVisits++;
23.         foreach (var cnn in node.Connections.OrderBy(x => x.Cost))
24.         {
25.             var childNode = cnn.ConnectedNode;
26.             if (childNode.Visited)
27.                 continue;
28.             if (childNode.MinCostToStart == null ||
29.                 node.MinCostToStart + cnn.Cost < childNode.MinCostToStart)
30.             {
31.                 childNode.MinCostToStart = node.MinCostToStart + cnn.Cost;
32.                 childNode.NearestToStart = node;
33.                 if (!prioQueue.Contains(childNode))
34.                     prioQueue.Add(childNode);
35.             }
36.         }
37.         node.Visited = true;
38.         if (node == End)
39.             return;

```

```

40.     } while (prioQueue.Any());
41. }

```

Kode Sumber 4.2 Implementasi algoritma A*

```

1.  public List<Node> GetShortestPathDijkstra()
2.  {
3.      DijkstraSearch();
4.      var shortestPath = new List<Node>();
5.      shortestPath.Add(End);
6.      BuildShortestPath(shortestPath, End);
7.      shortestPath.Reverse();
8.      return shortestPath;
9.  }
10.
11. private void BuildShortestPath(List<Node> list, Node node)
12. {
13.     if (node.NearestToStart == null)
14.         return;
15.     list.Add(node.NearestToStart);
16.     BuildShortestPath(list, node.NearestToStart);
17. }
18.
19. private void DijkstraSearch()
20. {
21.     Start.MinCostToStart = 0;
22.     var prioQueue = new List<Node>();
23.     prioQueue.Add(Start);
24.     do {
25.         prioQueue = prioQueue.OrderBy(x => x.MinCostToStart).ToList();
26.
27.         var node = prioQueue.First();
28.         prioQueue.Remove(node);
29.         foreach (var cnn in node.Connections.OrderBy(x => x.Cost))
30.         {
31.             var childNode = cnn.ConnectedNode;
32.             if (childNode.Visited)
33.                 continue;
34.             if (childNode.MinCostToStart == null ||
35.                 node.MinCostToStart + cnn.Cost < childNode.MinCostToSta
rt)
36.             {

```

```

36.         childNode.MinCostToStart = node.MinCostToStart + cnn.Co
    st;
37.         childNode.NearestToStart = node;
38.         if (!prioQueue.Contains(childNode))
39.             prioQueue.Add(childNode);
40.     }
41. }
42. node.Visited = true;
43. if (node == End)
44.     return;
45. } while (prioQueue.Any());
46. }

```

Kode Sumber 4.3 Implementasi algoritma Dijkstra

Setelah dilakukan implementasi menggunakan beberapa algoritma *shortest path*. Diputuskan algoritma yang dipakai adalah dijkstra dikarenakan memiliki akurasi tinggi dengan waktu tercepat dan memori yang tidak terlalu jauh berbeda.

Dijkstra yang dipakai telah disesuaikan dengan peraturan permainan Traffic Panic. *Code* akan dipanggil setiap kali kendaraan muncul. Implementasi dapat dilihat pada Kode Sumber 4.4.

```

1. public void GeneratePathTo(int x, int y, GameObject unit) {
2.     unit.GetComponent<Unit>().currentPath = null;
3.     Dictionary<Node, float> dist = new Dictionary<Node, float>();
4.     Dictionary<Node, Node> prev = new Dictionary<Node, Node>();
5.     List<Node> unvisited = new List<Node>();
6.     Node source = nodes[ unit.GetComponent<Unit>().tileX, unit.GetCompo
    nent<Unit>().tileY ];
7.     Node target = nodes[ x, y ];
8.     dist[source] = 0;
9.     prev[source] = null;
10.    foreach(Node v in nodes) {
11.        if(v != source) {
12.            dist[v] = Mathf.Infinity;
13.            prev[v] = null;
14.        }
15.        unvisited.Add(v);

```

```

16.     }
17.     while(unvisited.Count > 0) {
18.         Node u = null;
19.         foreach(Node possibleU in unvisited) {
20.             if(u == null || dist[possibleU] < dist[u]) {
21.                 u = possibleU;
22.             }
23.         }
24.         if(u == target) {
25.             break;
26.         }
27.         unvisited.Remove(u);
28.         foreach(Node v in u.neighbours) {
29.             float alt = dist[u] + u.DistanceTo(v);
30.             if( alt < dist[v] ) {
31.                 dist[v] = alt;
32.                 prev[v] = u;
33.             }
34.         }
35.     }
36.     if(prev[target] == null) {
37.         return;
38.     }
39.     List<Node> currentPath = new List<Node>();
40.     Node curr = target;
41.     while(curr != null) {
42.         currentPath.Add(curr);
43.         curr = prev[curr];
44.     }
45.     currentPath.Reverse();
46.     unit.GetComponent<Unit>().currentPath = currentPath;
47. }

```

Kode Sumber 4.4 Implementasi algoritma Dijkstra pada Unity

4.2.2. Implementasi Pergerakan Kendaraan

Setiap kendaraan akan bergerak menuju tujuan, sebagaimana yang telah dijelaskan dapat dilihat pada Kode Sumber 4.5.


```

1. void Update() {
2.     // advance pathing to next node
3.     if(Vector3.Distance(transform.position, map.TileCoordToWorldCoord(
        tileX, tileY )) < 0.03f){
4.         AdvancePathing();
5.
6.         // unoccupy node behind
7.         if(UNtileX != -1){
8.             if(Vector3.Distance(transform.position, map.TileCoordToWorldCoord(
                UNTileX, UNTileY )) >= 0.4f)
9.                 UnOccupying();
10.        }
11.        // move the unit
12.        if((( map.nodes[tileX,tileY].occupied == null && map.nodes[tileX,tileY].booked == false ) || map.nodes[tileX,tileY].occupied == gameObject.invisible) && mogok == 0) {
13.            Move();
14.        }
15.        else if(map.nodes[tileX,tileY].booked == false && map.nodes[tileX,tileY].occupied != null)
16.        {
17.            float plusTraffic = Random.Range(0.01f, unpatience);
18.            traffic += plusTraffic;
19.            happiness -= Random.Range(0.01f, friendliness/2);
20.            map.GetComponent<gameController>().traffic += plusTraffic;
21.        }
22.    }
23.
24.    void Move(){
25.        happiness += Random.Range(0.01f, friendliness);
26.
27.        Vector3 normalizeDirection = (map.TileCoordToWorldCoord( tileX, tileY ) - transform.position).normalized;
28.    }
29.
30.    void AdvancePathing() {
31.        if(currentPath==null)
32.            return;
33.
34.        transform.position = map.TileCoordToWorldCoord( tileX, tileY );
35.        UNTileX = tileX;
36.        UNTileY = tileY;
37.    }

```

```

38.     if(currentPath.Count == 1) {
39.         map.GetComponent<gameController>().traffic -= traffic/2;
40.         if(happiness > traffic && happiness > 0)
41.             map.GetComponent<gameController>().happiness += happiness -
traffic;
42.         PlayerPrefs.SetInt("coinLevel", PlayerPrefs.GetInt("coinLevel")
+ Random.Range(coinmin, coinmax));
43.         Destroy(this.gameObject);
44.         return;
45.     }
46.     tileX = currentPath[1].x;
47.     tileY = currentPath[1].y;
48.     currentPath.RemoveAt(0);
49. }
50. // unoccupy
51. void UnOccupying() {
52.     if(map.nodes[UNtileX,UNtileY].occupied == gameObject)
53.         map.nodes[UNtileX,UNtileY].occupied = null;
54. }

```

Kode Sumber 4.5 Pergerakan kendaraan

4.2.3. Implementasi Pengatur Lampu Lalu Lintas

Pemain dapat mengatur warna lampu lalu lintas dengan melakukan *tap* pada salah satu lampu lalu lintas sebagaimana yang telah dijelaskan dapat dilihat pada Kode Sumber 4.6.

```

1. void Update () {
2.     if(currentLight == "green") {
3.         map.nodes[x,y].booked = false;
4.         GetComponent<SpriteRenderer>().sprite = TLPic[0];
5.         map.nodes[x,y].occupied = null;
6.     }
7.     else if(currentLight == "yellow") {
8.         map.nodes[x,y].booked = false;
9.         GetComponent<SpriteRenderer>().sprite = TLPic[1];
10.        map.nodes[x,y].occupied = null;
11.    }
12.    else if(currentLight == "red") {
13.        map.nodes[x,y].booked = true;
14.        if(map.nodes[x,y].occupied == null)
15.            map.nodes[x,y].occupied = gameObject;

```

```

16.         GetComponent().sprite = TLPic[2];
17.     }
18. }
19.
20. void OnMouseDown()
21. {
22.     if(!crossroadHandler.broken)
23.         crossroadHandler.changeTo(id);
24. }

```

Kode Sumber 4.6 Pengatur Lampu Lalu Lintas

4.2.4. Implementasi Pengontrol Bangunan

Setiap waktu tertentu, kendaraan akan keluar dari bangunan dan/atau ujung jalan. Implementasi dapat dilihat pada Kode Sumber 4.7.

```

1. void Update () {
2.     if(homeStatus > 1)
3.     {
4.         timerObj.SetActive(true);
5.         System.TimeSpan timertemp = System.TimeSpan.FromSeconds(deb
uffTimer);
6.         string timeTemp = string.Format("{0:00}:{1:00}", timertemp.
Minutes, timertemp.Seconds);
7.         timerText.GetComponent<TextMesh>().text = timeTemp;
8.     }
9.     else{
10.        if(timerObj != null)
11.        {
12.            timerObj.SetActive(false);
13.        }
14.    }
15.
16.    if(timerPU > 0)
17.    {
18.        timerPU-= Time.deltaTime;
19.    }
20.    else if(timerPU <= 0 && homeStatus == 1)
21.    {
22.        homeStatus = 0;
23.    }
24.    if(homeStatus == 0)

```

```

25.     {
26.         for(int i=0; i < vehicleCount; i++ )
27.         {
28.             if(timer[i] > 0){
29.                 if(map.GetComponent<gameController>().freezePU)
30.                     timer[i] -= Time.deltaTime/4f;
31.                 else
32.                     timer[i] -= Time.deltaTime;
33.             }
34.             else {
35.                 GameObject[] temp = GameObject.FindGameObjectsWithT
ag("normalCar");
36.                 if(map.GetComponent<gameController>().maxCar > temp
.Length)
37.                     setter(i);
38.             }
39.         }
40.     }
41.     if(homeStatus > 1)
42.     {
43.         if(debuffTimer <= 0)
44.         {
45.             map.GetComponent<gameController>().gameLose();
46.         }
47.         if( !beingHandled )
48.         {
49.             StartCoroutine( timeManager() );
50.         }
51.     }
52. }
53.
54. void setter(int i)
55. {
56.     timer[i] = spawnTime[i];
57.     Vector3 spawnPlace = map.TileCoordToWorldCoord( spawnX, spawnY
);
58.     int carRandom = Random.Range(0,prefabsVehicle.Length);
59.     vehicle[i] = Instantiate(prefabsVehicle[carRandom], spawnPlace,
Quaternion.identity);
60. }

```

Kode Sumber 4.7 Pengontrol Bangunan

4.2.5. Implementasi Perhitungan Skor

Pengaturan skor yang mencakup segala hal yang berhubungan dengan *scoring* dalam permainan. Kode dapat dilihat di Kode Sumber 4.8.

```

1.     public void gameLose()
2.     {
3.         losePanel.SetActive(true);
4.         endGame = true;
5.         float timenya = timeLapse / 60;
6.         if(timeLapse/60 > 8)
7.         {
8.             timenya = 8;
9.         }
10.        float xpCount = (10f - timenya);
11.        xpCount *= (Mathf.Pow(2f, (float) level)) * ((float)stage/5f);
12.        xpCount += PlayerPrefs.GetFloat("xp");
13.    }
14.
15.    public void gameWin()
16.    {
17.        winPanel.SetActive(true);
18.        endGame = true;
19.        float timenya = timeLapse / 60;
20.        if(timeLapse/60 > 8)
21.        {
22.            timenya = 8;
23.        }
24.        float xpCount = ((10f - timenya) * ((trafficMax - traffic) / tr
afficMax));
25.        float increaseRate = (float) level * 1.5f + (float) stage/10f;
26.
27.        xpCount *= (Mathf.Pow(5f, (float) level)) * ((float)stage);
28.        xpCount += PlayerPrefs.GetFloat("xp");
29.    }

```

Kode Sumber 4.8 Perhitungan Skor

4.3. Implementasi Antarmuka

4.3.1. Implementasi *Main Menu*

Implementasi dari Main Menu mencakup tombol menuju ke *district selection* yaitu dengan menggunakan tombol ‘play’ , tombol setting yang memunculkan *pop-up setting game*, tombol Carpedia dengan *icon* mobil yang menuju ke *scene* Carpedia berisi penjelasan tiap mobil, tombol *leaderboard* yang menuju *scene* leaderboard online, tombol *achievement* dengan *icon* piala yang menuju *scene achievement* berisi kumpulan penghargaan yang telah dicapai pemain, dan tombol *shop* untuk menuju ke *scene* toko untuk membeli banyak hal. *Screenshot* dari *Main Menu* dapat dilihat pada Gambar 4.1.



Gambar 4.1 Implementasi tampilan *Main Menu*

4.3.2. Implementasi *District Selection*

Implementasi *District Selection* mencakup daftar *district* yang dapat dimainkan dan dapat diakses sebelum pemain memilih *stage*, serta terdapat pilihan untuk kembali ke main menu. *Screenshot* dari Level Menu dapat dilihat pada Gambar 4.2.

District yang tersedia adalah *town* dan *city* yang memiliki karakteristik dan tingkat kesulitan yang berbeda



Gambar 4.2 Implementasi tampilan *district selection*

4.3.3. Implementasi *Stage Selection*

Implementasi *stage selection* mencakup daftar *stage* dari *district* yang dipilih sebelumnya, serta terdapat pilihan untuk kembali ke *district selection*. *Screenshot* dari *gameplay* dapat dilihat pada Gambar 4.3.



Gambar 4.3 Implementasi tampilan *stage selection*

4.3.4. Implementasi *Gameplay Stage*

Implementasi *Gameplay Stage* mencakup komponen *stage* berupa jalan, kendaraan, bangunan, lampu lalu lintas, *happiness bar*, *panic bar*, *time*, *coin*, tombol *power-up*, dan tombol pause. *Screenshot* dari *gameplay* dapat dilihat pada Gambar 4.3.



Gambar 4.4 Implementasi Gameplay

BAB V

PENGUJIAN DAN EVALUASI

5.1. Lingkungan Pengujian

Lingkungan pelaksanaan uji coba meliputi perangkat keras dan perangkat lunak yang akan digunakan pada sistem ini. Spesifikasi perangkat keras dan perangkat lunak yang digunakan dalam rangka uji coba perangkat lunak ini dicantumkan pada Tabel 5.1.

Tabel 5.1 Lingkungan pengujian

Tipe perangkat	Spesifikasi
Perangkat Keras	Lenovo Vibe K-Note 4 CPU: Dual-core 2.6GHz Memori : 2.00 GB
Perangkat Lunak	Sistem Operasi Android 5.0

5.2. Pengujian Fungsionalitas

Untuk mengetahui kesesuaian keluaran dari tiap tahap dan langkah penggunaan fitur terhadap skenario yang dipersiapkan, maka dibutuhkan pengujian fungsionalitas.

Pengujian ini dilakukan untuk menguji apakah fungsionalitas yang diidentifikasi benar-benar diimplementasikan dan bekerja sebagaimana seharusnya. Pengujian juga dilakukan untuk mengetahui kesesuaian setiap tahapan atau langkah penggunaan fitur terhadap skenario yang dipersiapkan. Pengujian dilakukan dengan metode *black-box*.

5.2.1. Skenario Pengujian Fungsionalitas

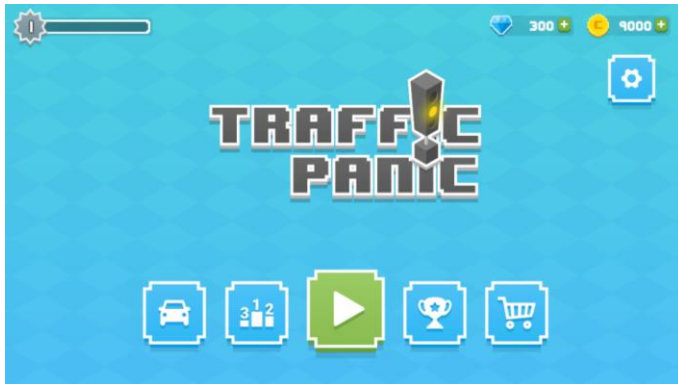
Skenario pengujian fungsionalitas digunakan untuk memberikan tahap-tahap dalam pengujian sistem. Skenario tertera pada Tabel 5.2.

Tabel 5.2 Pengujian aplikasi permainan

Pengujian	Hasil pengujian
Kondisi Awal	Pengguna berada pada layar Main Menu
Prosedur Pengujian	Pengguna masuk ke <i>District Selection</i> , <i>Stage Selection</i> , memilih <i>Stage</i> yang akan dimainkan, lalu memainkan permainan hingga selesai
Hasil yang diharapkan	Pengguna berhasil menyelesaikan permainan dan fungsionalitas permainan berjalan dengan lancar.
Hasil yang diperoleh	Pengguna berhasil menyelesaikan permainan dan fungsionalitas berjalan lancar.
Kesimpulan	Pengujian berhasil

5.2.1.1. Pengujian Main Menu sampai Stage Selection

Pengujian dimulai ketika pengguna telah masuk ke layar Main menu seperti yang terlihat pada Gambar 5.1. Pemain memilih menekan tombol ‘Play’ untuk masuk ke dalam layar *District Selection*.



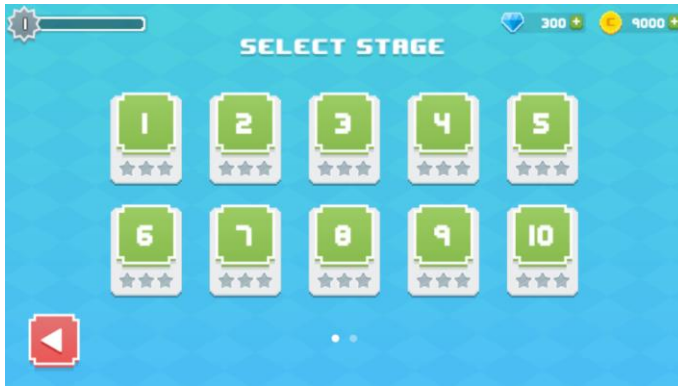
Gambar 5.1 Tampilan Main Menu

Setelah menekan tombol ‘Play’ maka sistem menampilkan layar *District Selection* seperti yang terlihat pada Gambar 5.2.



Gambar 5.2 Tampilan District Selection

Setelah itu, pemain memilih *district* dan melanjutkan ke tampilan *Stage Selection* seperti yang terlihat pada Gambar 5.3. Selain itu di tampilan *district selection* juga terdapat tombol ‘back’ yang dapat digunakan untuk kembali ke tampilan *Main Menu*



Gambar 5.3 Tampilan *Stage Selection*

Setelah masuk ke tampilan *stage selection*, pemain dapat memilih *stage* yang ingin dimainkan dengan memilih salah satu *stage*. Selain itu di tampilan *stage selection* juga terdapat tombol ‘back’ yang dapat digunakan untuk kembali ke tampilan *District Selection*.

Setelah melakukan pengujian, sistem berhasil masuk ke *Game Menu* dan melakukan load sesuai *stage* yang dipilih. Selain itu sistem juga berhasil kembali ke *Main Menu* ketika pengguna menekan tombol ‘back’. Sehingga dapat disimpulkan bahwa pengujian untuk layar *Main Menu* sampai *Stage Selection* berhasil.

5.2.1.2. Pengujian Penggantian Warna Lampu Lali Lintas

Pengujian dimulai ketika pemain telah masuk ke *gameplay stage*. Pemain dapat melakukan *tap* pada lampu lalu lintas. Pengujian *tap* pada lampu lalu lintas sehingga lampu indikator berubah sesuai peraturan permainan. sehingga dapat disimpulkan bahwa pengujian untuk penggantian warna lampu lalu lintas berhasil. Berikut hasil pengujian ini dapat dilihat pada Gambar 5.4.



Gambar 5.4 Tampilan penggantian warna lampu lalu lintas

5.2.1.3. Pengujian Penggunaan *Power-Up*

Pengujian dimulai ketika pemain telah masuk ke *gameplay stage*. Pemain dapat menggunakan semua *power-up* yang ada. Setelah diuji disimpulkan bahwa *power-up* berhasil digunakan. Berikut hasil dari pengujian ini dapat dilihat pada Gambar 5.5.



Gambar 5.5 Tampilan penggunaann *power-up*

5.2.1.4. Pengujian Aturan Jalan Kendaraan

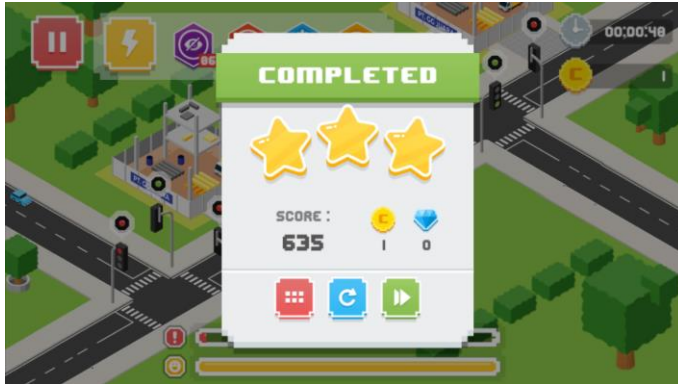
Pengujian dimulai ketika pemain telah masuk ke *gameplay stage*. Untuk kondisi awalnya tidak ada kendaraan di jalan. Lalu seiring berjalannya waktu, akan muncul kendaraan dari rumah dan ujung jalan. Kendaraan munjul, berjalan melalui jalan sesuai *shortest path* menuju titik tujuan menandakan kendaraan telah berhasil berjalan mengikuti aturan permainan. Berikut hasil dari pengujian ini dapat dilihat pada gambar 5.6.



Gambar 5.6 Kendaraan berjalan sesuai aturan permainan

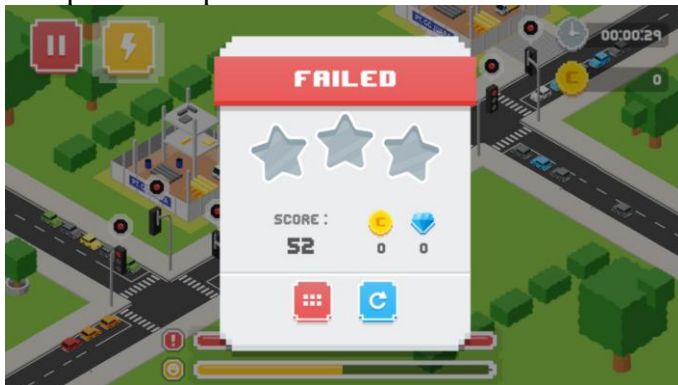
5.2.1.5. Pengujian Menang dan Kalah

Pengujian ini dimulai dengan menyelesaikan *stage* permainan. Saat *happiness bar* penuh sebelum *panic bar* penuh. Tampilan pemain menang dapat dilihat pada Gambar 5.7.



Gambar 5.7 Tampilan saat pemain menang

Pemain akan dinyatakan kalah saat *panic bar* penuh sebelum *happiness bar* penuh. Tampilan pengujian pemain kalah dapat dilihat pada Gambar 5.8.



Gambar 5.8 Tampilan saat pemain kalah

5.2.2. Hasil Pengujian Fungsionalitas

Hasil uji fungsionalitas yang telah dilakukan berdasarkan skenario sebelumnya, menunjukkan bahwa semua fungsionalitas permainan berjalan dengan baik dan sesuai dengan sebagaimana mestinya yang telah dibuat pada

tahap perancangan. Hasil uji fungsionalitas dapat dilihat pada Tabel 5.3.

Tabel 5.3 Hasil pengujian fungsionalitas

No	Pengujian	Hasil Pengujian
1	<i>Main Menu</i> hingga <i>Stage Selection</i>	Berhasil
2	Penggantian warna lalu lintas	Berhasil
3	Penggunaan <i>power-up</i>	Berhasil
4	Aturan jalan kendaraan	Berhasil
5	Menang dan Kalah	Berhasil

5.3. Pengujian Pengguna

Pengujian pengguna adalah untuk mendapatkan informasi mengenai pendapat pengguna mengenai *game* ‘Traffic Panic’. Pengujian dilakukan ke sepuluh pengguna.

5.3.1. Skenario Pengujian Pengguna

Pengujian dilakukan oleh pengguna (pemain) sebanyak 10 anak usia sekolah dasar dengan memainkan *game* ‘Traffic Panic’. Sebelum bermain, pemain akan dijelaskan mengenai Undang- undang nomor 22 tahun 2009 tentang lalu lintas dan angkutan jalan. Pemain lalu memainkan seluruh *stage* pada *district* pertama. Kemudian pemain diberikan kuisioner dengan beberapa pertanyaan. Pertanyaan yang diberikan meliputi pendapatnya mengenai *game* ‘Traffic Panic’, dan masukan untuk pengembangan aplikasi permainan lebih lanjut.

5.3.2. Hasil Pengujian Pengguna

Setelah dilakukan pengujian oleh pengguna, didapatkan beberapa informasi. Pengguna berpendapat bahwa dari segi penggunaan aplikasi permainan, ‘Traffic Panic’ sudah baik. Aplikasi permainan mudah untuk

dimengerti dan sudah nyaman untuk digunakan. *Gameplay* permainan juga menarik dan permainan sudah dapat berjalan sebagaimana mestinya. Informasi *stage* yang dicoba pemain ditunjukkan pada Tabel 5.4.

Tabel 5.4 Hasil Pengujian Pengguna

Pengguna	<i>District - Stage</i> yang dicoba	Jumlah Dimainkan	Jumlah Menang
1	1-1	10	10
2	1-2	10	10
3	1-3	10	10
4	1-4	10	10
5	1-5	10	10
6	1-6	10	9
7	1-7	10	8
8	1-8	10	9
9	1-9	10	10
10	1-10	10	10
11	1-11	10	9
12	1-12	10	8
13	1-13	10	8
14	1-14	10	7
15	1-15	10	8
16	1-16	10	7
17	1-17	10	5
18	1-18	10	4

Hasil dari kuisioner yang diisi para pengguna dapat dilihat pada gambar Tabel 5.5. dan Tabel 5.6.

Tabel 5.5 Hasil Kuisioner Pengujian Pengguna

No	usia	Jenis kela- min	Keme- narikan UI	UI UX mudah dime- ngerti ?	Peme- rataan tingkat ke- sulitan	Edukasi tentang peraturan Lalu- lintas	Sesuai undang- undang nomor 22 tahun 2009
1	11	L	5	5	5	5	5
2	11	P	5	4	3	5	5
3	9	P	5	4	2	5	4
4	9	L	5	5	4	5	4
5	10	P	5	5	3	5	5
6	8	L	5	4	2	5	3
7	12	L	5	4	4	5	4
8	9	L	5	4	4	5	4
9	11	L	5	5	5	5	5
10	9	L	5	5	4	5	4

Tabel 5.6 Rangkuman Hasil Kuisioner Pengujian Pengguna

Penilaian	Total nilai	Presentase	Total	Total Presentase
Kemenarikan UI	50	100%	224	89,6%
UI/UX mudah dimengerti	45	90%		
Pemerataan tingkat kesulitan	36	72%		
Edukasi peraturan lalu lintas	50	100%		
Kesesuaian dengan undang- undang nomor 22 tahun 2009	43	86%		

5.4. Pengujian Algoritma *Shortest Path*

Pengujian bertujuan untuk menguji ketepatan algoritma *shortest path* dengan game ‘traffic panic’. Aspek yang dinilai adalah akurasi atau ketepatan, *run-time*. Algoritma yang di uji meliputi A*, IDA* dan Dijkstra.

5.4.1. Skenario Pengujian Algoritma *Shortest Path*

Pengujian dilakukan dengan *generate map* menyerupai game ‘traffic panic’ dan menggunakan peraturan permainan. Setelah *generate map*, Masing- masing algoritma kan dijalankan dan dicoba sebanyak 10 kali dengan *start node* dan *end node* random, sehingga mendapatkan hasil yang dapat dibandingkan.

5.4.2. Hasil Pengujian Algoritma *Shortest Path*

Setelah dilakukan pengujian masing masing program didapatkan hasil Algoritma A* memiliki rata- rata *run-time* 0.018 detik, dengan rata- rata jumlah *node* yang dikunjungi adalah 42. Algoritma IDA* memiliki rata- rata *run-time* 0.578 detik, dengan rata- rata jumlah *node* yang dikunjungi adalah 58, dan algoritma Dijkstra memiliki rata- rata *run-time* 0.013 detik, dengan rata- rata jumlah *node* yang dikunjungi adalah 37. Sedangkan ketiga algoritma memiliki akurasi yang sama, yaitu 100%. Informasi dan perbandingan hasil pengujian ditunjukkan pada Tabel 5.7.

Tabel 5.7 Hasil Pengujian Algoritma *Shortest Path*

Algoritma	Rata- rata <i>run-time</i> (s)	Rata- rata jumlah <i>node</i> yang dikunjungi	Rata- rata akurasi
A*	0.0 18	42	100%
IDA*	0.578	58	100%
Dijkstra	0.013	37	100%

5.5. Evaluasi Pengujian

Hasil Pengujian Fungsionalitas menunjukkan bahwa semua fungsionalitas permainan berjalan dengan baik dan sesuai dengan sebagaimana mestinya yang telah dibuat pada tahap perancangan dan tidak diperlukanya perubahan karena semua fungsionalitas dalam aplikasi berjalan sesuai seharusnya. Tetapi tidak menutup untuk adanya pengembangan fungsionalitas kedepanya.

Sedangkan untuk pengujian pengguna menunjukan bahwa setelah *game* diujikan ke target pengguna, menunjukan *game* sudah teruji dan mendapat *feedback* yang baik, yaitu 89,6%.

Pengguna paling menyukai *game* ‘Traffic Panic’ dari segi tampilan. Terbukti 100% nilai didapatkan dari *feedback* kemenarikan UI dan 90% *feedback* didapatkan dari UI/UX yang *user friendly*.

Dari segi edukasi, ‘Traffic Panic’ juga mendapat *feedback* yang sangat baik yaitu 100% pengguna memberi nilai 5 (sangat setuju) saat ditanyai mengenai *in-game education*. *Game* juga sudah menjawab rumusan masalah mengenai “Bagaimana merancang permainan simulasi pengaturan lalu lintas kota berdasarkan peraturan perundang-undangan di Indonesia sesuai undang undang nomor 22 tahun 2009 tentang lalu lintas dan angkutan jalan” dengan mendapat *feedback* 86%.

Untuk segi tingkat kesulitan, *game* ‘Traffic Panic’ masih perlu ditingkatkan lagi karena hanya mendapat *feedback* pengguna sebesar 72%.

Pengujian Algoritma menunjukan bahwa algoritma paling efektif untuk pencarian *shortest path game* ‘traffic panic’ adalah Dijkstra dengan akurasi 100% dan waktu *run-time* hanya 0.013 detik mengalahkan algoritma A* dengan

akurasi 100% dan waktu *run-time* 0.018 detik, dan algoritma IDA* dengan akurasi 100% dan waktu *run-time* 0.578 detik. Dengan demikian diputuskan algoritma yang akan dipakai untuk penyelesaian masalah *shortest path* dalam game ‘Traffic Panic’ adalah Dijkstra.

[Halaman ini sengaja dikosongkan]

BAB VI

KESIMPULAN DAN SARAN

Bab ini membahas mengenai kesimpulan yang dapat diambil dari hasil uji coba yang telah dilakukan sebagai jawaban dari rumusan masalah yang dikemukakan. Selain kesimpulan, juga terdapat saran yang ditujukan untuk pengembangan penelitian lebih lanjut.

6.1. Kesimpulan

Dari proses penerapan *game* Traffic Panic dapat diambil kesimpulan sebagai berikut:

1. 'Traffic Panic' berhasil dirancang dengan baik menerapkan pergerakan karakter permainan yang diadopsi dari permainan yang mirip dan menggunakan *gameplay* aturan permainan yang dirancang sendiri oleh penulis.
2. Aturan perancangan pencarian jalan terpendek menggunakan algoritma Dijkstra dengan menyesuaikan *gameplay* terbukti lebih efektif dengan akurasi 100% dan waktu *run-time* hanya 0.013 detik mengalahkan algoritma A* dengan akurasi 100% dan waktu *run-time* 0.018 detik, dan algoritma IDA* dengan akurasi 100% dan waktu *run-time* 0.578 detik
3. Penerapan algoritma Dijkstra dalam pencarian jalan terpendek dalam *game* dapat diterapkan dalam *game engine*, dan permainan dapat berjalan sebagaimana mestinya dalam perangkat mobile.
4. Traffic Panic terbukti menyampaikan nilai- nilai positif mengenai peraturan perlalulintasan karena 10 ari 10 hasil kuisisioner yang diisi responden yang menyatakan bahwa mereka mendapat ilmu baru

tentang perlalulintasan dari *game* ‘Traffic Panic’
Sesuai undang- undang nomor 22 tahun 2009.

6.2. Saran

Saran yang diberikan terkait pengembangan pada Tugas Akhir ini berdasarkan perancangan, implementasi, dan uji coba adalah:

1. Menambah parameter tingkat kesulitan dalam penyusunan *puzzle stage*.
2. Menambah *option* bahasa indonesia
3. Memperbanyak jumlah pengujian terhadap setiap *stage*.
4. Menambah fitur *mode* permainan *Endless* dan penambahan macam *power-up*.
5. Penambahan *skill tree* untuk meng-*upgrade power-up*.
6. Perbanyak *stage* dan *district*.

DAFTAR PUSTAKA

- [1] Ijsselsteijn, Wijanand et al. (2008). Measuring the Experience of Digital Game Enjoyment. Proceedings of Measuring Behavior 2008, 6th International Conference on Methods and Techniques in Behavioral Research. Maastricht, Netherlands, August 26-29, 2008.
- [2] Dix, Alan et al. (2004). Human-Computer Interaction 3rd. England: Pearson Education Limited.
- [3] Unity. "CREATE GAMES, CONNECT WITH YOUR AUDIENCE, AND ACHIEVE SUCCESS". Unity Technologies, 2017. [Online]. Available: <http://unity3d.com/unity>. [Diakses 2 September 2017].
- [4] Kemppainen, Jaakko. (2014). Designing a Knowledge Based Puzzle Game. Finlandia: Aalto University.
- [5] Buck, Jamis. (2015). *Maze* for Programmers. United States of America: The Pragmatic Programmers.
- [6] Android Official Website (2018) — "Android | Official Website", <<http://www.android.com/>>.
- [7] G.B. Dantzig, All shortest routes in a graph, Theory of Graphs, Int. Symp., Rome 1966(Dunod, Paris, 1967) pp. 91–92.
- [8] Stanford Theory Group, "Amit's A* page", <http://theory.stanford.edu/~amitp/GameProgramming/AStarComparison.html>, accessed October 12, 2010.
- [9] *Korf, Richard (1985). "Depth-first Iterative-Deepening: An Optimal Admissible Tree Search". Artificial Intelligence. 27: 97–109*

- [10] Feng - sheng, X. U. (2004). Algorithm for Finding the Shortest Paths [J]. Computer Applications, 5.

BIODATA PENULIS



Penulis lahir di Kediri, 19 April 1998, merupakan anak ketiga dari empat bersaudara. Dalam menjalani pendidikan semasa hidup, penulis menempuh pendidikan di TK Bayangkari Kediri, SD plus Ar-Rahman Banaran Kediri, MTS Negeri 2 Kediri, SMA Negeri 1 Kediri dan S1 Departemen Informatika Institut Teknologi Sepuluh Nopember (ITS) pada rumpun Interaksi Grafika dan Seni (IGS).

Selama menjadi mahasiswa, penulis ikut dalam Himpunan Mahasiswa Teknik Komputer Informatika ITS, dalam departemen Ristik sebagai staf magang pada tahun kedua, staf Schematics 2016 di biro NPC. Dan dilanjutkan pada Schematics 2017. Penulis telah menghasilkan beberapa karya aplikasi permainan diantaranya: 'Ummi', 'KABUUUR', 'Blue Diamond', 'Froggy Battle' dan 'Traffic Panic'. Berkat kegemarannya di ranah *game* development, penulis mendapatkan beberapa pencapaian dan penghargaan dalam kompetisi *game* developing. Penulis mengikuti beberapa kompetisi seperti ajang Gemastik 10 yang diselenggarakan di Universitas Indonesia berhasil menjadi finalis dengan membawakan *game* 'Froggy Battle', mendapatkan penghargaan *Best UI/UX* pada ajang Google Developer Student Club Hackaton di Institut Teknologi Sepuluh Nopember dengan membawakan *game* 'Ummi', mendapatkan penghargaan *best game* pada Game Jam 1 2018

dengan membawakan *game* ‘Blue Diamond’, mendapatkan penghargaan *favorite game* pada Game Jam 3 2018 dengan membawakan *game* ‘KABUUUR’ dan mendapatkan medali emas kategori aplikasi permainan pada ajang Gemastik 11 di Institut Teknologi Sepuluh Nopember dengan membawakan *game* ‘Traffic Panic’.

LAMPIRAN A



Gambar A.6.1 *Stage 1-1*



Gambar A.6.2 *Stage 1-2*



Gambar A.6.3 Stage 1-3



Gambar A.6.4 Stage 1-4



Gambar A.6.5 Stage 1-5



Gambar A.6.6 Stage 1-6



Gambar A.6.7 Stage 1-7



Gambar A.6.8 Stage 1-8



Gambar A.6.9 Stage 1-9



Gambar A.6.10 Stage 1-10



Gambar A.6.11 *Stage 1-11*



Gambar A.6.12 *Stage 1-12*



Gambar A.6.13 Stage 1-13



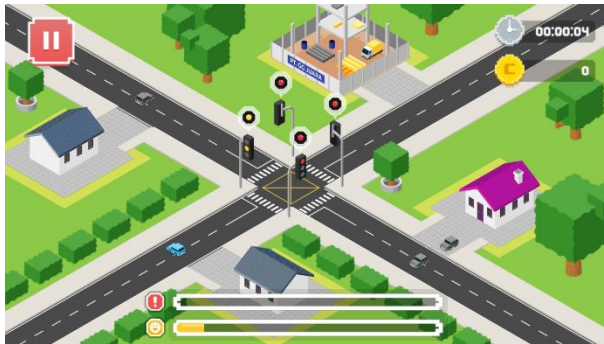
Gambar A.6.14 Stage 1-14



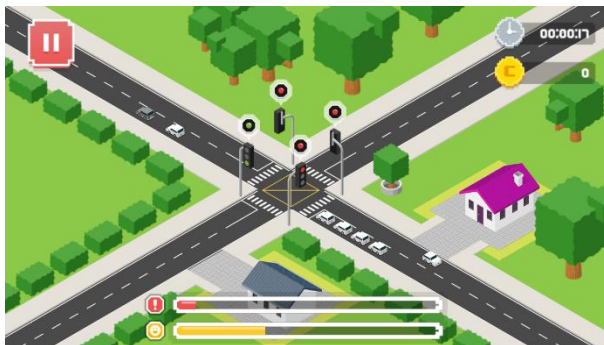
Gambar A.6.15 *Stage 1-15*



Gambar A.6.16 *Stage 1-16*



Gambar A.6.17 Stage 1-17



Gambar A.6.18 Stage 1-18



Gambar A.6.19 *District 2*

LAMPIRAN B

jenis kelamin *

☒ L

☐ P

Barapa usia anda? *

11

Gambar dan tampilan game 'Traffic Panic' menarik

	1	2	3	4	5	
sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	sangat setuju

Tampilan 'Traffic Panic' Mudah dimengerti

	1	2	3	4	5	
sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	sangat setuju

Setiap level di 'Traffic Panic' mempunyai tingkat kesulitan yang pas

	1	2	3	4	5	
sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	sangat setuju

Saya belajar hal baru mengenai perlatu-lintasan kota saat bermain 'Traffic Panic'

	1	2	3	4	5	
sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	sangat setuju

Game 'Traffic Panic' sesuai undang undang yang tadi dijelaskan

	1	2	3	4	5	
sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	sangat setuju

saran pengembangan game 'Traffic Panic'

perbanyak level, macam powerup kurang, uang sedikit

Gambar B.1 *Form* pengujian pengguna 1

jenis kelamin *

☐ L

☒ P

Barapa usia anda? *

11

Gambar dan tampilan game 'Traffic Panic' menarik

	1	2	3	4	5	
sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	sangat setuju

Tampilan 'Traffic Panic' Mudah dimengerti

	1	2	3	4	5	
sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	sangat setuju

Setiap level di 'Traffic Panic' mempunyai tingkat kesulitan yang pas

	1	2	3	4	5	
sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	sangat setuju

Saya belajar hal baru mengenai perlatu-lintasan kota saat bermain 'Traffic Panic'

	1	2	3	4	5	
sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	sangat setuju

Game 'Traffic Panic' sesuai undang undang yang tadi dijelaskan

	1	2	3	4	5	
sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	sangat setuju

saran pengembangan game 'Traffic Panic'

susah kalo gapakek powerup

Gambar B.2 Form pengujian pengguna 2

jenis kelamin *

☐ L

☒ P

Barapa usia anda? *

9

Gambar dan tampilan game 'Traffic Panic' menarik

	1	2	3	4	5	
sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	sangat setuju

Tampilan 'Traffic Panic' Mudah dimengerti

	1	2	3	4	5	
sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	sangat setuju

Setiap level di 'Traffic Panic' mempunyai tingkat kesulitan yang pas

	1	2	3	4	5	
sangat tidak setuju	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	sangat setuju

Saya belajar hal baru mengenai per lalu-lintasan kota saat bermain 'Traffic Panic'

	1	2	3	4	5	
sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	sangat setuju

Game 'Traffic Panic' sesuai undang undang yang tadi dijelaskan

	1	2	3	4	5	
sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	sangat setuju

saran pengembangan game 'Traffic Panic'

susah

Gambar B.3 Form pengujian pengguna 3

jenis kelamin *

☒ L

☐ P

Barapa usia anda? *

9

Gambar dan tampilan game 'Traffic Panic' menarik

	1	2	3	4	5	
sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	sangat setuju

Tampilan 'Traffic Panic' Mudah dimengerti

	1	2	3	4	5	
sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	sangat setuju

Setiap level di 'Traffic Panic' mempunyai tingkat kesulitan yang pas

	1	2	3	4	5	
sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	sangat setuju

Saya belajar hal baru mengenai per lalu-lintasan kota saat bermain 'Traffic Panic'

	1	2	3	4	5	
sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	sangat setuju

Game 'Traffic Panic' sesuai undang undang yang tadi dijelaskan

	1	2	3	4	5	
sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	sangat setuju

saran pengembangan game 'Traffic Panic'

.....

Gambar B.4 Form pengujian pengguna 4

jenis kelamin *

☐ L

☒ P

Barapa usia anda? *

10

Gambar dan tampilan game 'Traffic Panic' menarik

	1	2	3	4	5	
sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	sangat setuju

Tampilan 'Traffic Panic' Mudah dimengerti

	1	2	3	4	5	
sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	sangat setuju

Setiap level di 'Traffic Panic' mempunyai tingkat kesulitan yang pas

	1	2	3	4	5	
sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	sangat setuju

Saya belajar hal baru mengenai perlatu-lintasan kota saat bermain 'Traffic Panic'

	1	2	3	4	5	
sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	sangat setuju

Game 'Traffic Panic' sesuai undang undang yang tadi dijelaskan

	1	2	3	4	5	
sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	sangat setuju

saran pengembangan game 'Traffic Panic'

suka mobil pink

Gambar B.5 Form pengujian pengguna 5

jenis kelamin *

☒ L

☐ P

Barapa usia anda? *

9

Gambar dan tampilan game 'Traffic Panic' menarik

	1	2	3	4	5	
sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	sangat setuju

Tampilan 'Traffic Panic' Mudah dimengerti

	1	2	3	4	5	
sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	sangat setuju

Setiap level di 'Traffic Panic' mempunyai tingkat kesulitan yang pas

	1	2	3	4	5	
sangat tidak setuju	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	sangat setuju

Saya belajar hal baru mengenai perlatu-lintasan kota saat bermain 'Traffic Panic'

	1	2	3	4	5	
sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	sangat setuju

Game 'Traffic Panic' sesuai undang undang yang tadi dijelaskan

	1	2	3	4	5	
sangat tidak setuju	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	sangat setuju

saran pengembangan game 'Traffic Panic'

tidak bisa bahasa inggris, diganti bahasa indonesia

Gambar B.6 Form pengujian pengguna 6

jenis kelamin *

☒ L

☐ P

Barapa usia anda? *

12

Gambar dan tampilan game 'Traffic Panic' menarik

	1	2	3	4	5	
sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	sangat setuju

Tampilan 'Traffic Panic' Mudah dimengerti

	1	2	3	4	5	
sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	sangat setuju

Setiap level di 'Traffic Panic' mempunyai tingkat kesulitan yang pas

	1	2	3	4	5	
sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	sangat setuju

Saya belajar hal baru mengenai per lalu-lintasan kota saat bermain 'Traffic Panic'

	1	2	3	4	5	
sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	sangat setuju

Game 'Traffic Panic' sesuai undang undang yang tadi dijelaskan

	1	2	3	4	5	
sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	sangat setuju

saran pengembangan game 'Traffic Panic'

diamond susah didapatkan

Gambar B.7 Form pengujian pengguna 7

jenis kelamin *

☒ L

☐ P

Barapa usia anda? *

9

Gambar dan tampilan game 'Traffic Panic' menarik

	1	2	3	4	5	
sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	sangat setuju

Tampilan 'Traffic Panic' Mudah dimengerti

	1	2	3	4	5	
sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	sangat setuju

Setiap level di 'Traffic Panic' mempunyai tingkat kesulitan yang pas

	1	2	3	4	5	
sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	sangat setuju

Saya belajar hal baru mengenai perlatu-lintasan kota saat bermain 'Traffic Panic'

	1	2	3	4	5	
sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	sangat setuju

Game 'Traffic Panic' sesuai undang undang yang tadi dijelaskan

	1	2	3	4	5	
sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	sangat setuju

saran pengembangan game 'Traffic Panic'

suka saat nabrak dan macet

Gambar B.8 Form pengujian pengguna 8

jenis kelamin *

☒ L

☐ P

Barapa usia anda? *

11

Gambar dan tampilan game 'Traffic Panic' menarik

	1	2	3	4	5	
sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	sangat setuju

Tampilan 'Traffic Panic' Mudah dimengerti

	1	2	3	4	5	
sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	sangat setuju

Setiap level di 'Traffic Panic' mempunyai tingkat kesulitan yang pas

	1	2	3	4	5	
sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	sangat setuju

Saya belajar hal baru mengenai perlatu-lintasan kota saat bermain 'Traffic Panic'

	1	2	3	4	5	
sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	sangat setuju

Game 'Traffic Panic' sesuai undang undang yang tadi dijelaskan

	1	2	3	4	5	
sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	sangat setuju

saran pengembangan game 'Traffic Panic'

terlalu cepet selesai

Gambar B.9 Form pengujian pengguna 9

jenis kelamin *

☒ L

☐ P

Barapa usia anda? *

10

Gambar dan tampilan game 'Traffic Panic' menarik

	1	2	3	4	5	
sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	sangat setuju

Tampilan 'Traffic Panic' Mudah dimengerti

	1	2	3	4	5	
sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	sangat setuju

Setiap level di 'Traffic Panic' mempunyai tingkat kesulitan yang pas

	1	2	3	4	5	
sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	sangat setuju

Saya belajar hal baru mengenai perlatu-lintasan kota saat bermain 'Traffic Panic'

	1	2	3	4	5	
sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	sangat setuju

Game 'Traffic Panic' sesuai undang undang yang tadi dijelaskan

	1	2	3	4	5	
sangat tidak setuju	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	sangat setuju

saran pengembangan game 'Traffic Panic'

yang city susah pol

Gambar B.10 Form pengujian pengguna 10